

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2006-107514
(43)Date of publication of application : 20.04.2006

(51)Int.Cl. G06F 15/80 (2006.01)
G06F 9/52 (2006.01)
G06F 9/54 (2006.01)

(21)Application number : 2005-292708 (71)Applicant : SONY COMPUTER ENTERTAINMENT INC
(22)Date of filing : 05.10.2005 (72)Inventor : SUZUKI MASAKAZU YAMAZAKI TAKESHI

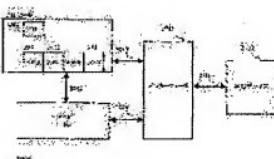
(30)Priority Priority number : 2004 959635 Priority date : 05.10.2004 Priority country : US

(54) SYSTEM AND DEVICE WHICH HAVE INTERFACE DEVICE WHICH CAN PERFORM DATA COMMUNICATION WITH EXTERNAL DEVICE

(57) Abstract:

PROBLEM TO BE SOLVED: To provide an architecture which enables various members on a network to share data and applications among them without an additional load in computing.

SOLUTION: A system is provided, which has a processing element (PE), an input/output (I/O) interface device, and a shared memory. The PE has at least one processing unit (PU) and also has one or more additional processing devices (APU). At least one of the APUs executes an I/O function by reading from or writing to an external device connected to the I/O interface device 2915. Data is exchanged between the APU and the I/O interface device 2915 by using a data level synchronization mechanism through the shared memory.



(10) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2006-107514

(P2006-107514A)

(13) 公開日 平成18年4月20日(2006.4.20)

(51) Int.Cl.

G06F 15/80 (2006.01)
 G06F 9/32 (2006.01)
 G06F 9/54 (2006.01)

F 1

G06F 15/80
 G06F 9/46 475C
 G06F 9/46 480Z

テーマコード(参考)

(21) 出願番号 特願2005-292708 (P2005-292708)
 (22) 出願日 平成17年10月5日 (2005.10.5)
 (31) 優先権主張番号 10/959,635
 (32) 優先日 平成16年10月5日 (2004.10.5)
 (33) 優先権主張国 米国(US)

(特許庁注:以下のものは登録商標)
 1. JAVA
 2. イーサネット

(71) 出願人 395015319
 株式会社ソニー・コンピュータエンタテインメント
 東京都港区南青山二丁目6番21号
 (74) 代理人 100099324
 弁理士 鈴木 正剛
 100108604
 弁理士 村松 義人
 (74) 代理人 100111615
 弁理士 庄野 良太
 (72) 発明者 鈴置 雅一
 東京都港区南青山二丁目6番21号 株式会社ソニー・コンピュータエンタテインメント内

最終頁に続く

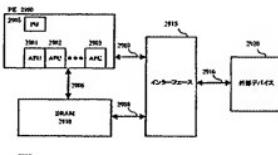
(54) 【発明の名称】外部デバイスとデータ通信可能なインターフェイスデバイスを有するシステム及び装置

(57) 【要約】

【課題】計算上の負担が付加されることなく、ネットワークの様々なメンバ間でのデータとアプリケーションの共用可能なアーキテクチャを提供する。

【解決手段】処理エレメント(PE)、入出力(I/O)インターフェースデバイス及び共有メモリを有するシステムが提供される。PEは、処理ユニット(PU)を少なくとも一つ備え、かつ、一つ以上の付加処理装置(APU)を備える。APUの少なくとも一つは、I/Oインターフェースデバイス2915に接続された外部デバイスに対して読み出しあるいは書き込みを行うことでI/O機能を実行する。データは、APUとI/Oインターフェースデバイス2915との間で、データレベル同期メカニズムを用いて共有メモリを通じて交換される。

【選択図】図29



【特許請求の範囲】**【請求項 1】**

外部デバイスへのデータ通信と外部デバイスからのデータ通信とを行うように動作可能なインターフェースデバイスと、

各々が前記データの格納をするように動作可能な、複数のメモリ・ロケーションを持つメモリとを含み、

前記インターフェースデバイスと前記メモリのうちの少なくとも 1 つが、前記メモリ・ロケーションの対応するメモリ・ロケーションと関連付けられた状態情報を格納するよう に動作可能であって、前記状態情報には、第 1 フィールドとアドレス・フィールドとが含まれ、この第 1 フィールドとアドレス・フィールドとは、与えられたメモリ・ロケーションに対して、前記関連する状態情報の前記第 1 フィールドの値が第 1 の値と等しくかつ前記関連する状態情報の前記アドレス・フィールドの値が第 2 の値と等しい場合に、前記メモリ・ロケーションへの書き込みオペレーションによって、前記メモリ・ロケーションに現在格納されているデータが、前記第 2 の値によって示されるアドレスへの書き込みが行われる、システム。

【請求項 2】

前記アドレスは前記外部デバイスと関連付けられている、請求項 1 記載のシステム。

【請求項 3】

前記アドレスは、前記インターフェースと通信を行うプロセッサと関連付けられているメモリ・ロケーションと関連付けられている、請求項 1 記載のシステム。

20

【請求項 4】

データアクセス用に前記メモリに結合され、かつ前記インターフェースと結合されているプロセッサを含み、前記プロセッサは前記外部デバイスと関連付けられた要求を発行するよう に動作可能である、請求項 1 又は 2 又は 3 記載のシステム。

【請求項 5】

少なくとも 1 つの前記要求は、前記外部デバイスからのデータの読み出しに対するものである、請求項 4 記載のシステム。

【請求項 6】

少なくとも 1 つの前記要求は、前記外部デバイスへのデータの書き込みに対するものである、請求項 4 記載のシステム。

30

【請求項 7】

前記メモリは、前記プロセッサと前記インターフェースデバイスとの間に前記要求を送るよう に動作可能である、請求項 4 又は 5 又は 6 記載のシステム。

【請求項 8】

前記外部デバイスは、前記インターフェースデバイスと結合されている、請求項 1 ~ 7 のいずれかに記載のシステム。

【請求項 9】

前記インターフェースデバイスは、DMA 伝送を用いて、前記外部デバイスと前記メモリとの間でのデータ通信を行う、請求項 1 ~ 8 のいずれかに記載のシステム。

40

【請求項 10】

複数のメモリ・ロケーションを持つメモリと、

少なくとも 1 つの第 1 のタイプと第 2 のタイプの要求を発行するよう に動作可能で前記メモリと結合されたプロセッサと、

インターフェースデバイスと、を含み、前記インターフェースデバイスは、前記第 1 のタイプの要求を受信したことに応答して、前記外部デバイスから前記メモリへデータを伝送し、前記メモリへ前記データを格納するよう に動作可能であり、かつ、前記インターフェースデバイスは前記第 2 のタイプの要求を受信したことに応答して、前記メモリのストレージから前記外部デバイスへデータを伝送するよう に動作可能であって、

前記メモリの少なくとも 1 つのメモリ・ロケーションと、前記インターフェースデバイスとは、ブロッキング状態を含む、複数の異なるメモリの状態をサポートし、このサー

50

トでは、前記ブロッキング状態にある与えられたメモリ・ロケーションへの書き込みオペレーションによって、現在、その中に格納されているデータが、前記与えられたメモリ・ロケーションに関連付けられたアドレスに書き込まれる、システム。

【請求項11】

前記アドレス値は前記外部デバイスを表す、請求項10記載のシステム。

【請求項12】

前記プロセッサと関連付けられたローカル・メモリを更に含み、前記アドレス値は前記ローカル・メモリのメモリ・ロケーションを表わす、請求項10又は11記載のシステム。

【請求項13】

前記第1のタイプの前記要求は、前記外部デバイスからのデータの読み出しに対するものである、請求項10又は11又は12記載のシステム。

10

【請求項14】

前記第2のタイプの前記要求は、前記外部デバイスからのデータの書き込みに対するものである、請求項10又は11又は12記載のシステム。

【請求項15】

前記外部デバイスは前記インターフェースデバイスと結合されている、請求項10～14のいずれかに記載のシステム。

【請求項16】

前記インターフェースデバイスは、DMA伝送を用いて、前記外部デバイスと前記メモリとの間でのデータ通信を行う、請求項10～15のいずれかに記載のシステム。

【請求項17】

前記プロセッサは、プロセシングユニット(PU)と、前記要求を発行するように動作可能な、少なくとも1つの付加処理ユニット(APU)とを含む、請求項10～16のいずれかに記載のシステム。

【請求項18】

前記インターフェースデバイスは、前記メモリ・ロケーションへのアクセスを制御するための保護テーブルを含む、請求項17記載のシステム。

【請求項19】

前記PUは、前記保護テーブルの値を初期化する、請求項18記載のシステム。

30

【請求項20】

外部装置とプロセッサ間を相互接続する装置であって、

保護テーブルを含み、前記保護テーブルは前記プロセッサによってアクセス可能な共用メモリの領域を示す情報を格納するように動作可能であって、

伝送コントローラを含み、前記伝送コントローラはプロセッサからの要求に応答して、前記外部デバイスと前記共用メモリの前記アクセス可能な領域のうちの少なくとも1つとの間のデータ伝送を制御するものであって、かつ、

伝送バスを含み、前記伝送バスは前記伝送コントローラによる制御の下で、前記外部デバイスと、前記共用メモリの前記少なくとも1つのアクセス可能な領域との間に前記データを伝送するように動作可能であり、

前記伝送コントローラは、メモリ・ロケーションの最新のメモリ状態によって、前記少なくとも1つのアクセス可能な領域のメモリ・ロケーションと外部デバイスとの間のデータ伝送を同期して制御するよう動作可能であり、この制御では、最新のメモリ状態がブロッキング状態である場合、メモリ・ロケーションへの書き込みオペレーションにより、現在メモリ・ロケーションに格納されているデータが、前記メモリ・ロケーションに関連付けられたアドレスへ書き込まれる、装置。

40

【請求項21】

インターフェースデバイスを介して、プロセッサと外部デバイスとの間にデータ伝送を行う方法であって、

前記外部デバイスと前記プロセッサに対してローカルであるメモリとの間で、前記デー

50

タを、複数のメモリ・ロケーションを持つ共用メモリを介して前記プロセッサへ伝送し、前記共用メモリは、ブロッキング状態を含む複数の異なるメモリ状態をサポートし、前記ブロッキング状態にある与えられたメモリ・ロケーションへの書き込みオペレーションにより、現在前記メモリ・ロケーションに格納されているデータが、前記所望のメモリ・ロケーションと関連付けられたアドレス値へ書き込まれる、方法。

【請求項 2 2】

前記プロセッサは第1プロセッサであり、更に、

第2プロセッサからインターフェースデバイスへ、インターフェースデバイスの保護テーブルにおいて用いるための値を送り、前記値は前記共用メモリの一部へのアクセスを制御する、請求項 2 1 記載の方法。 10

【請求項 2 3】

前記外部デバイスは、

外部デバイスを通じて、伝送されるデータを要求するコマンドを受信し、

前記コマンドに関連付けられたアドレス値が有効であるかどうかを判断するために、保護テーブルの値をチェックし、

アドレス値が有効である場合は、前記初期化ステップを行う、請求項 2 1 又は 2 2 記載の方法。

【請求項 2 4】

前記関連するアドレス値は前記外部デバイスを示す、請求項 2 1 又は 2 2 又は 2 3 記載の方法。 20

【請求項 2 5】

前記関連するアドレス値は前記プロセッサと関連付けられたローカル・メモリを示す、請求項 2 1 又は 2 2 又は 2 3 記載の方法。

【請求項 2 6】

デバイスであって、

要求や応答を複数の要求／応答チャネルを介して転送するための第1バスを含み、各要求／応答チャネルはプロセッサと関連付けられており、各要求／応答チャネルは、前記関連付けられたプロセッサと外部デバイスとの間にデータ通信用の要求を転送するものであり、

メモリとのデータ通信を行う第2バスを含み、前記第2バスはデータレベル同期を用いて前記データの通信時に用いられるものであり、前記メモリは複数のメモリ・ロケーションを含み、前記各メモリ・ロケーションは、ブロッキング状態を含む複数の異なるメモリ状態をサポートするものであって、前記ブロッキング状態にある与えられたメモリ・ロケーションへのデータの書き込みオペレーションにより、現在前記与えられたメモリ・ロケーションに格納されているデータが、前記与えられたメモリ・ロケーションと関連付けられているアドレス値へ書き込まれる、デバイス。

【請求項 2 7】

前記第1バスは、前記プロセッサによりアクセス可能なメモリの一部を示す前記値を、保護テーブルで用いられるように転送する、請求項 2 6 記載のデバイス。

【発明の詳細な説明】

40

【技術分野】

【00001】

本発明は、同時係属中であり、本発明の譲受人に譲渡された以下の米国特許出願、2001年3月22日出願の米国特許出願第09/816,004号「広帯域ネットワークのコンピュータ・アーキテクチャおよびソフトウェア・セル」、2001年3月22日出願の米国特許出願第09/815,554号「広帯域ネットワーク用のコンピュータ・アーキテクチャのデータ同期システムおよび方法」、2001年3月22日出願の米国特許出願09/816,020号「広帯域ネットワーク用のコンピュータ・アーキテクチャのメモリ保護システムおよび方法」、2001年3月22日出願の米国特許出願第09/815,558号「広帯域ネットワーク用のコンピュータ・アーキテクチャの資源占有システ

50

ムおよび方法」、および2001年3月22日出願の米国特許出願第09/816,752号「広帯域ネットワーク用のコンピュータ・アーキテクチャの処理モジュール」の一部継続出願であり、その全てをここに援用する。

【背景技術】

【0002】

本発明はコンピュータ・プロセッサ用アーキテクチャとコンピュータ・ネットワークに関する、より詳細には広帯域環境におけるコンピュータ・プロセッサ及びコンピュータ・ネットワーク用アーキテクチャに関する。

【0003】

コンピュータ及び現今のコンピュータ・ネットワークのコンピューティング・デバイス(オフィスのネットワークで使用されるローカル・エリア・ネットワーク(LAN)やインターネットなどのようなグローバルネットワークなど)の計算用デバイスは、スタンド・アローン型の計算用として主として設計されていた。コンピュータ・ネットワークを介するデータとアプリケーション・プログラム(“アプリケーション”)の共用は、これらのコンピュータ及びコンピューティング・デバイスの主要な設計目標ではなかった。これらのコンピュータとコンピューティング・デバイスは、様々な異なるメーカー(モトローラ、インテル、テキサス・インスツルメント、ソニーなど)により製造された広範囲の異なるタイプのプロセッサを用いて一般に設計されたものである。これらのプロセッサの各々はそれ自身の特定の命令セットと命令セット・アーキテクチャ(I SA : instruction set architecture)を有している。すなわち、それ自身の特定のセットのアセンブリ言語命令と、これらの命令を実行する主演算デバイスと記憶デバイスのための構造とを有する。従って、プログラマは各プロセッサの命令セットとI SAとを理解してこれらのプロセッサ用のアプリケーション書くことを要求される。今日のコンピュータ・ネットワーク上でのコンピュータとコンピューティング・デバイスに異なった種類が混在していることから、データとアプリケーションの共用及びその処理は複雑になっている。さらに、この複数種が混在する環境に対する調整を行うために、多くの場合、同じアプリケーションであっても複数のバージョンを用意することが必要となっている。

【発明の開示】

【発明が解決しようとする課題】

【0004】

グローバルネットワーク、特にインターネットに接続されたタイプのコンピュータやコンピューティング・デバイスは広範囲に及ぶ。パーソナルコンピュータ(PC)やサーバーに加えて、これらのコンピューティング・デバイスの中には携帯電話、移動用コンピュータ、個人用情報機器(PDA : personal digital assistant)、セット・トップ・ボックス、デジタルテレビ並びにその他のデバイスが含まれる。コンピュータやコンピューティング・デバイスにおいて異種製品が混在する中でのデータやアプリケーションを共用することに起因して、重要な問題が発生している。

【0005】

これらの問題を解決するためのいくつかの手法が試みられてきた。これらの手法の中には特に、優れたインターフェースと複雑なプログラミング手法が含まれる。多くの場合、これらの解決方法では、処理パワーの実質的増加の実現が要求される。また、これらの解決方法では、多くの場合、アプリケーションの処理に必要な時間と、ネットワークを介するデータ伝送に必要な時間とが実質的に増加してしまうという結果が生じる。

【0006】

一般に、データは対応のアプリケーションとは別々に、インターネットを介して伝送される。この手法では、アプリケーションに対応した各セットの伝送データにアプリケーション自体をも送る必要はなくなっている。従ってこの手法により、必要とされる帯域幅の量は最小化されるものの、ユーザには不満の原因となることも多々ある。つまり、クライアント側のコンピュータでは、この伝送データを利用するための適正なアプリケーション、あるいは最新のアプリケーション入手できない事態も生じうる。またこの手法では、

10

20

30

40

50

ネットワーク上のプロセッサによって用いられている複数の異種ISAと命令セットに対応して、各アプリケーション毎にバージョンの異なる複数のアプリケーションを用意することが要求される。

【0007】

Javaモデルではこの問題の解決が試みられている。このモデルでは厳しいセキュリティ・プロトコルに準拠する小さなアプリケーション（“アプレット”（applet））が用いられている。アプレットはネットワークを介してサーバー側コンピュータから送信され、クライアント側コンピュータ（“クライアント”）により実行される。異なるISAを使用しているクライアントごとに、同じアプレットであっても異なるバージョンを送信するという事態を避ける必要があるため、全てのJavaアプレットはクライアント側のJava仮想マシン上で実行される。Java仮想マシンとは、JavaISAとJava命令セットを持つコンピュータをエミュレートするソフトウェアである。しかしながらこのソフトウェアはクライアント側のISAとクライアント側の命令セットにより実行される。クライアント側ではISAと命令セットが各々異なるが、与えられるJava仮想マシンのバージョンは1つである。従って、複数の各アプレットに異なるバージョンを用意する必要はない。各クライアントでは、当該クライアントにおけるISAと命令セットに対応した適切なJava仮想マシンだけをダウンロードすれば、全てのJavaアプレットを実行できる。

【0008】

各々の異なるISA命令セットに対して異なるバージョンのアプリケーションを書かなければならぬという課題は解決されているものの、Javaの処理モデルでは、クライアント側のコンピュータに対してソフトウェアの追加層が要求される。ソフトウェアのこの追加層のためにプロセッサの処理速度は著しく低下する。この速度の低下は、リアルタイムのマルチメディア・アプリケーションについて特に著しい。また、ダウンロードされたJavaアプレットの中には、ウイルス、処理上の誤動作などが含まれている可能性がある。これらのウイルスと誤動作は、クライアントのデータベースの破損やその他の損害の原因となる可能性がある。Javaモデルで用いられているセキュリティ用プロトコルでは、“サンンドボックス”（Javaアプレットがそれ以上はデータを書き込むことができない、クライアント側のメモリ内のスペース）というソフトウェアを設けることによりこの問題の解決が試みられているといえ、このソフトウェア駆動型セキュリティ・モデルは多くの場合、その実行時に不安定な状態になり、より多くの処理を必要とする。

【0009】

リアルタイムのマルチメディア・ネットワーク用アプリケーションがますます重要なものになりつつある。これらのネットワーク用アプリケーションは非常に高速な処理が要求される。将来、そのようなアプリケーション用として、毎秒何千メガビットものデータが必要になるかもしれない。ネットワークの現今のアーキテクチャ、及び、特にインターネットのアーキテクチャ、並びにJavaモデルなどで現在実施されているプログラミング・モデルでこのような処理速度に到達することは非常に難しい。

【0010】

従って、新しいコンピュータ・アーキテクチャと、コンピュータ・ネットワーク用の新しいアーキテクチャと、新しいプログラミング・モデルとが求められている。この新しいアーキテクチャとプログラミング・モデルによって計算上の負担が付加されることなく、ネットワークの様々なメンバー間でのデータとアプリケーションの共用という問題が解決されることが望ましい。この新たなコンピュータ・アーキテクチャとプログラミング・モデルとにより、ネットワークのメンバー間でのアプリケーションとデータの共用時に生じるセキュリティ上の問題も解決されることが望ましい。

【課題を解決するための手段】

【0011】

広帯域ネットワークを介して高速処理を行うコンピュータ・アーキテクチャが提供される。特に、及び、本発明によれば、システムにはデータ通信用のインターフェースデバイ

10

20

30

40

50

スと、データ格納用のメモリとが含まれ、メモリには少なくとも1つのメモリ・ロケーションと、それに関連する状態情報とが含まれる。状態情報には第1フィールドとアドレス・フィールドとが含まれ、第1フィールド値が第1の値に等しく、また、アドレス・フィールド値が第2の値に等しい場合、統いて行われる、メモリ・ロケーションに関連付けられたデータの書き込みによって、その中に格納されているデータが、アドレス・フィールド値により示されるアドレスへ書き込まれることになる。

【発明を実施するための最良の形態】

【0012】

本発明の一実施形態においては、システム構成にはプロセッサ・エレメント（P E）と、入力／出力（I / O）インターフェースデバイスと、共用メモリとが含まれる。P Eにはさらに、少なくとも1つの処理ユニット（P U）と、1つ以上の付加処理ユニット（A P U : attached processing unit）とが含まれる。少なくとも1つのA P Uは、I / Oインターフェースデバイスと結合されている外部デバイスからのデータの読出しと外部デバイスへのデータの書き込みを行うことにより、I / O機能を行う。データはA P UとI / Oインターフェースデバイスとの間で、共用メモリを介して、データレベル同期機構を用いて交換される。特に、共用メモリにはデータ格納用の少なくとも1つのメモリ・ロケーションと関連付けられた、少なくとも1つの状態情報ロケーションが含まれ、この状態情報ロケーションには第1フィールドとアドレス・フィールドが含まれる。第1フィールド値が第1の値に等しく、また、アドレス・フィールド値が第2の値に等しい場合、統いて行われる、メモリ・ロケーションに関連付けられたデータの書き込みによって、その中に格納されているデータが、アドレス・フィールド値により示されるアドレスへ書き込まれることになる。

10

【0013】

図1に、本発明によるコンピュータ・システム101のアーキテクチャ全体を示す。

【0014】

この図に例示されているように、システム101にはネットワーク104が含まれ、複数のコンピュータとコンピューティング・デバイスがこのネットワークと接続されている。ネットワーク104の例として、LAN、インターネットなどのグローバルネットワーク、又は他のコンピュータ・ネットワークが挙げられる。

20

【0015】

ネットワーク104に接続されているコンピュータとコンピューティング・デバイス（ネットワークの“メンバー”）の中には、クライアント側コンピュータ106、サーバー側コンピュータ108、個人用情報機器（P D A : personal digital assistant）、デジタルテレビ（D T V : digital television）112、及びその他の有線又は無線コンピュータとコンピューティング・デバイスなどが含まれる。ネットワーク104のメンバーにより用いられるプロセッサは、同じ共通のコンピューティング・モジュールから構成される。またこれらのプロセッサは、好適には、I S Aが全て同じで、同じ命令セットに従って処理を実行する。個々のプロセッサ内に含まれるモジュールの数は、そのプロセッサが必要とする処理パワーにより決められる。

30

【0016】

例えば、システム101のサーバー108はクライアント106より多いデータ及びアプリケーション処理を実行するので、サーバー108はクライアント106よりも多いコンピューティング・モジュールを含むことになる。一方、P D A 110では、最低量の処理しか実行されない。従って、P D A 110には最小の数のコンピューティング・モジュールしか含まれない。D T V 112はクライアント106とサーバー108間の処理レベルを実行する。従ってD T V 112にはクライアント106とサーバー108間のいくつかのコンピューティング・モジュールが含まれる。以下に解説するように、各コンピューティング・モジュールの中には、処理用コントローラと、ネットワーク104を介して伝送されるデータ及びアプリケーションの並列処理を実行する複数の同一処理ユニットとが含まれる。

40

50

【0017】

システム101がこのように均質な構成を有することから、アダプタリティ、処理速度、及び処理効率が改善される。システム101の各メンバーが、同じコンピューティング・モジュールの1つ以上（又はコンピューティング・モジュールの一部）を用いて処理を実行するので、データ及びアプリケーションの実際の処理をどのコンピュータ又はコンピューティング・デバイスで実行するかは重要ではなくなる。更に、個々のアプリケーション及びデータの処理は、ネットワークのメンバー間で分担することができる。システム全体を通じて、システム101が処理したデータ及びアプリケーションを含むセルを一意的に識別することにより、この処理がどこで行われたかに関わらず、処理を要求したコンピュータ又はコンピューティング・デバイスへその処理結果を伝送することが可能になる。この処理を実行するモジュールが共通の構造と共通のISAとを有するので、プロセッサ間の互換性を達成するためのソフトウエアの追加層の計算上の負荷が回避される。このアーキテクチャとプログラミング・モデルにより、リアルタイムのマルチメディア・アプリケーションなどの実行に必要な処理速度が改善される。

【0018】

システム101により改善された処理速度と効率という利点を更に利用するために、このシステムにより処理されるデータ及びアプリケーションは、一意的に識別される、それぞれフォーマットが同じであるソフトウエア・セル102へとパッケージ化される。各ソフトウエア・セル102はデータ及びアプリケーションの双方を含む、あるいは含み得る。各ソフトウエア・セルはまた、ネットワーク104とシステム101全体の中でセルをグローバルに識別するIDが含まれる。ソフトウエア・セルのこの構造的均一性と、ネットワークの中でのソフトウエア・セルの一意の識別によって、ネットワークの任意のコンピュータ又はコンピューティング・デバイスでのアプリケーションの処理が改善される。例えば、クライアント106はソフトウエア・セル102の作成を行なうこともできるが、クライアント106の処理能力は限られていることから、このソフトウエア・セルをサーバー108へ伝送して、処理してもらうこともできる。従って、ソフトウエア・セルはネットワーク104全体を移動してネットワーク上での処理用リソース可用性に基づく処理を行うことが可能となる。

【0019】

また、システム101のプロセッサとソフトウエア・セルが均質な構造を有することから、今日の異質なネットワークの混在という問題の多くを防ぐことができる。例えば、任意の命令セットを用いる任意のISA上でも、アプリケーションの処理を許容しようとする非効率なプログラミングモデル（Javaの仮想マシンのような仮想マシンなど）が回避される。従って、システム101は今日のネットワークよりもはるかに効率的、かつ、はるかに効率的に、広帯域処理の実現が可能になる。

【0020】

ネットワーク104の全てのメンバーのための基本となる処理用モジュールはプロセッサ・エレメント（PE）である。図2にPEの構造を例示する。この図に示すように、PE201は処理ユニット（PU）203、DMAC205、複数の付加処理ユニット（APU）、すなわち、APU207、APU209、APU211、APU213、APU215、APU217、APU219、APU221、を具備する。ローカルPEバス223はAPUと、DMAC205と、PU203との間でデータとアプリケーションを伝送する。ローカルPEバス223は従来のアーキテクチャなどを備えていてもよいし、又はパケット交換式ネットワークとして実装されてもよい。パケット交換式ネットワークとして実装される場合は、より多くのハードウエアが必要となり、その一方で利用可能な帯域幅が増加する。

【0021】

PEはデジタル論理回路を実装する様々な方法を用いて構成可能である。しかしながら、PE201は好適にはシリコン基板上の相補型金属酸化膜半導体（CMOS：complementary metal oxide semiconductor）を用いている単一の集積回路として構成される。基

10

20

30

40

50

板用代替材料の中には、ガリウム砒素、ガリウムアルミニウム砒素、及び多種多様のドーバントを用いるその他のいわゆるIII-I-B化合物が含まれる。またPE201は超伝導材料(高速單一磁束量子(RSFQ: rapid single-flux-quantum)論理回路など)を用いて実装されることもできる。

【0022】

PE201は高帯域メモリ接続部227を介して、ダイナミック・ランダム・アクセス・メモリ(DRAM)225と密接に関連している。DRAM225はPE201用メイン・メモリとして機能する。DRAM225は好適にはダイナミック・ランダム・アクセス・メモリであることが望ましいとはい、DRAM225は他の手段、例えば、スタティック・ランダム・アクセス・メモリ(SRAM)として、磁気ランダム・アクセス・メモリ(MRAM)、光メモリ又はホログラフィ・メモリなどを用いてDRAM225を実装することも出来る。DMAC205によってDRAM225と、PE201のAPUとPUとの間のデータ伝送が改善される。以下に更に説明するように、DMAC205によって、各APUに対するDRAM225内の排他的領域が指定されるが、この排他的領域の中へはAPUだけしかデータの書き込みができず、また、APUだけしかこの排他的領域からのデータ読み出しを行うことができない。また、この排他的領域は“サンドボックス”と呼ばれる。

【0023】

PU203は、データ及びアプリケーションのスタンド・アローン型処理が可能な、標準的プロセッサなどであってよい。作動時に、PU203はAPUによってデータ及びアプリケーションの処理のスケジュール管理と全般的管理を行なう。APUは好適には单一命令、複数データ(SIMD:single instruction, multiple data)プロセッサであることが望ましい。PU203の制御によって、APUは並列的かつ独立に、これらのデータ及びアプリケーションの処理を実行する。DMAC205は、共用DRAM225に格納されているデータとアプリケーションへのPU203とAPUによるアクセス制御を行う。PE201は好適には8個のAPUを含むことが望ましいとはい、必要とする処理パワーに応じて、PE内での数より多少上下する個数のAPUを用いてもよい。PU203とAPUの一部あるいは全ては、同じハードウェア構造、及び／又は機能を有することができる。個々のプロセッサは必要に応じて、ソフトウェアによって制御を行う、又は制御されるプロセッサとして構成される。例えば図3では、PE201は同じアーキテクチャを持つ9個のプロセッサを含むことができる。9個のプロセッサのうちの1つのプロセッサは、制御を行なうプロセッサ(例:PU203)と呼ばれ、残りのプロセッサは制御されるプロセッサと呼ばれる(APU207、209、211、213、215、217、219、212など)。また、PE201のようないくつかのPEを結合(まとめてパッケージ化)して処理パワーの改善を図ることができる。

【0024】

例えば、図3に示すように、一つ以上のチップ・パッケージなどの中に4つのPEをパッケージ化してネットワーク104のメンバー用の单一プロセッサを形成してもよい。この構成は広帯域エンジン(BE)と呼ばれる。図3に示すように、BE301には4つのPE(PE303、PE305、PE307、及びPE309)が含まれる。これらのPE間の通信は、BEバス311を介して行われる。広帯域メモリ接続部313によって共用DRAM315とこれらのPE間の通信が行われる。BEバス311の代わりに、BE301のPE間の通信は、DRAM315とこのメモリ接続部を介して行なうことができる。

【0025】

入力／出力(I/O)インターフェース317と外部バス319とは、広帯域エンジン301とネットワーク104の他のメンバー間で通信を行なう。BE301の各PEは、PEのAPUによって行われるアプリケーションとデータの並列的かつ独立した処理と同様の並列的かつ独立した方法で、データとアプリケーションの処理を実行する。

【0026】

10

20

30

40

50

図4にAPUの構造を例示する。APU402にはローカル・メモリ406、レジスタ410、4つの浮動小数点ユニット412、及び4つの整数演算ユニット414が含まれる。しかし、ここでもまた必要とする処理パワーに応じて、4個より多少上下する個数の浮動小数点演算ユニット412と整数演算ユニット414を用いてもよい。1つの好ましい実施形態では、ローカル・メモリ406には128キロバイトの記憶容量が含まれ、レジスタ410の容量は 128×128 ビットである。浮動小数点ユニット412は、毎秒320億浮動小数点演算(32GLPOS)で好適に作動し、整数ユニット414は毎秒320億回の演算速度(32GOP)で好適に作動する。

【0027】

ローカル・メモリ402はキャッシュ・メモリではない。ローカル・メモリ402は好適にはSRAMとして構成されることが望ましい。APUに対するキャッシュ・コヒーレンシー、つまりキャッシュの整合性のサポートは不要である。PUは当該PUで開始されるダイレクト・メモリ・アクセス(DMA)をサポートするために、キャッシュの整合性が要求される場合もある。しかし、APU又は外部デバイスからの、及び外部デバイスへのアクセスに対するキャッシュの整合性は不要である。

【0028】

APU402には更に、APUへ及びAPUからアプリケーションとデータを伝送するためのバス404が含まれる。1つの好ましい実施形態では、このバスは1024ビットの幅を持つ。APU402には更に内部バス408と、420と、418とが含まれる。1つの好ましい実施形態では、バス408は256ビットの幅を持ち、ローカル・メモリ406とレジスタ410間で通信を行う。バス420と418とは、それぞれ、レジスタ410と浮動小数点演算ユニット412との間、及びレジスタ410と整数ユニット414との間で通信を行う。1つの好ましい実施形態では、レジスタ410から浮動小数点演算ユニット412又は整数ユニットへのバス418の幅は318ビットであり、浮動小数点演算ユニット412又は整数演算ユニット414からレジスタ410へのバス418と420の幅は128ビットである。浮動小数点演算ユニット412又は整数演算ユニット414への幅より広い、レジスタ410から浮動小数点演算ユニットまたは整数演算ユニットへの上記バスの広い幅によって、レジスタ410からのより広いデータ・フローが処理中に許容される。各計算には最大3ワードが必要になる。しかし、各計算結果は一般に1ワードだけである。

【0029】

図5-10は、ネットワーク104のメンバーのプロセッサのモジュラー構造を更に例示する図である。例えば、図5に示すように、1つのプロセッサには単一のPE502を含むことができる。上述のように、このPEには一般に、PUと、DMACと、8個のAPUとが含まれる。各APUにはローカル・ストレージ(LS)が含まれる。一方、プロセッサはビジュアライザ(VS)505の構造を有する。図5に示すように、VS505はPUS12と、DMAC514と、4つのAPU(APU516、APU118、APU520、APU522)とを有する。PEのその他の4つのAPUによって通常占領されるチップ・パッケージ内のスペースは、この場合、ピクセル・エンジン508、画像用キャッシュ510、及びラウンドローラ(CRTC)504によって占領される。PE502又はVS505に求められる通信速度に応じて、チップ・パッケージの中に光インターフェース506が含まれる場合もある。

【0030】

この標準化されたモジュラー構造を用いて、多数の他のプロセッサが、容易にかつ効率的に構成されうる。例えば、図6に示すプロセッサは、2つのチップ・パッケージ(BE)を備えるチップ・パッケージ602と、4つのVSを含むチップ・パッケージ604)を有する。入出力部(I/O)606によって、チップ・パッケージ602のBEとネットワーク104との間にインターフェースが設けられる。バス608はチップ・パッケージ602とチップ・パッケージ604との間で通信を行う。入出力プロセッサ(IOP)610によってデータ・フローが制御され、I/O606への、またはI/O606からの

10

20

30

40

50

入出力が行われる。I/O 606は特定用途向集積回路(A S I C : application specific integrated circuit)として製造が可能である。V Sからの出力はビデオ信号612である。

【0031】

図7に、ネットワーク104のその他のメンバーへ超高速通信を行なう2つの光インターフェース704と706などを備えたB E 702用のチップ・パッケージ(またはローカルに接続された、その他のチップ・パッケージ)を例示する。B E 702はネットワーク104のサーバーなどとして機能することができます。

【0032】

図8のチップ・パッケージは2つのP E 802と804、及び2つのV S 806と808を有する。I/O 810はチップ・パッケージとネットワーク104との間にインターフェースを与える。チップ・パッケージからの出力はビデオ信号1である。この構成は画像処理用ワークステーションなどとして機能することができます。

【0033】

図9に更に別の構成を例示する。この構成は、図8に例示されている構成の処理パワーの1/2を含む。2つのP Eの代わりに、1つのP E 902が設けられ、2つのV Sの代わりに1つのV S 904が設けられる。I/O 906は図8に例示されているI/Oの帯域幅の1/2の帯域幅を有する。しかしこのようなプロセッサは、画像処理用ワークステーションとしても機能することができます。

【0034】

最後の構成を図10に示す。このプロセッサは單一のV S 1002とI/O 1004だけから構成される。この構成はP D Aなどとして機能することができます。

【0035】

図11Aにネットワーク104のプロセッサのチップ・パッケージの中への光インターフェースの統合を例示する図を示す。これらの光インターフェースによって、光信号は電気信号に変換され、電気信号は光信号へ変換される。また、これらの光インターフェースは、ガリウム砒素、アルミニウムガリウム砒素、ゲルマニウム、及びその他の元素や化合物などを含む様々な材料から構成される。この図に示すように、光インターフェース1104と1106はB E 1102のチップ・パッケージ上に組み立てられる。B Eバス1108はB E 1102のP E、すなわち、P E 1110、P E 1112、P E 1114、P E 1116及びこれらの光インターフェースとの間で通信を行う。光インターフェース1104には2つのポート(ポート1118とポート1120)が含まれ、また光インターフェース1106には2つのポート(ポート1122とポート1124)が含まれる。ポート1118、1120、1122及び1124は光導波路1126、1128、1130、1132とそれぞれ接続される。光信号は光インターフェース1104と1106のポートを介して、これらの光導波路の中を通り、B E 1102へ、及びB E 1102から伝送される。

【0036】

このような光導波路と各B Eの4つの光ポートとを用いて、様々な構成において複数のB Eをまとめて接続してもよい。例えば図11Bに示すように、このような光ポートを介して2つまたはそれ以上のB E(B E 1152、B E 1154、B E 1156など)を直列に接続することができる。この例では、B E 1152の光インターフェース1166は、その光ポートを介してB E 1154の光インターフェース1160の光ポートと接続される。同様に、B E 1154の光インターフェース1162の光ポートは、B E 1156の光インターフェース1164の光ポートと接続される。

【0037】

図11Cにマトリクス構成を例示する。この構成では、各B Eの光インターフェースは2つの他のB Eと接続される。この図に示すように、B E 1172の光インターフェース1188の光ポートの中の1つが、B E 1176の光インターフェース1182の光ポートと接続される。光インターフェース1188のもう一方の光ポートは、B E 1178の

10

20

30

40

50

50

光インターフェース 1 1 8 4 の光ポートと接続される。同様に、B E 1 1 7 4 の光インターフェース 1 1 9 0 の 1 つの中の光ポートは、B E 1 1 7 8 の光インターフェース 1 1 8 4 のもう一方の光ポートと接続される。光インターフェース 1 1 9 0 のもう一方の光ポートは、B E 1 1 8 0 の光インターフェース 1 1 8 6 の光ポートと接続される。このマトリクス構成は他の B E に対しても同様に拡張することができる。

【0038】

シリアル構成かマトリクス構成のいずれかを用いて、任意の所望のサイズとパワーからなるネットワーク 1 0 4 用プロセッサ構成が可能になる。言うまでもなく、B E の光インターフェースに対して、あるいは B E より P E 数が上下するプロセッサに対して追加ポートを加えて他の構成を形成してもよい。

10

【0039】

図 1 2 A に B E の D R A M に対する制御システムと構造を示す。同様の制御システムと構造が、別のサイズを持ち、多少異なる数の P E を含むプロセッサの中で用いられる。この図に示すように、クロスバ交換機によって、B E 1 2 0 1 を備える 4 つの P E からなる各 D M A C 1 2 1 0 が 8 つのバンク・コントロール 1 2 0 6 と接続される。各バンク・コントロール 1 2 0 6 によって D R A M 1 2 0 4 の 8 つのバンク 1 2 0 8 (4 つだけしか図示されていない) が制御される。したがって、D R A M 1 2 0 4 は合計で 6 4 のバンクを具備することになる。好ましい実施形態では、D R A M 1 2 0 4 は 6 4 メガバイトの容量を持ち、各バンクは 1 メガバイトの容量を持つ。各バンク内の最小のアドレス指定可能な単位は、この好ましい実施形態では 1 0 2 4 ピットのブロックである。

20

【0040】

B E 1 2 0 1 にはスイッチ・ユニット 1 2 1 2 も含まれる。スイッチ・ユニット 1 2 1 2 により、B E 1 2 0 1 に直接に接続されている B E の他の A P U の、D R A M 1 2 0 4 へのアクセスが可能となる。したがって、第 2 の B E を第 1 の B E と直接に接続することが可能となり、さらに、各 B E の各 A P U は、A P U が通常アクセス可能なメモリ・ロケーション数の 2 倍のアドレス指定を行うことが可能となる。スイッチ・ユニット 1 2 1 2 のようなスイッチ・ユニットを介して、第 1 の B E の D R A M から第 2 の B E の D R A M へのデータの直接読み出し、または、第 2 の B E の D R A M から第 1 の B E の D R A M へのデータの直接書き込みを行なうことが可能となる。

【0041】

例えば、図 1 2 B に示すように、このような書き込みを行なうために、第 1 の B E の A P U (B E 1 2 2 2 の A P U 1 2 2 0 など) によって、第 2 の B E の D R A M (通常の場合のような B E 1 2 2 2 の B E 1 2 2 2 の D R A M 1 2 2 4 ではなく、B E 1 2 2 6 の D R A M 1 2 2 8 など) のメモリ・ロケーションへの書き込みコマンドが出される。B E 1 2 2 2 の D M A C 1 2 3 0 は、クロスバ交換機 1 2 2 1 を介してバンク・コントロール 1 2 3 4 へ書き込みコマンドを送り、バンク・コントロール 1 2 3 4 はバンク・コントロール 1 2 3 4 と接続された外部ポート 1 2 3 2 へコマンドを伝送する。B E 1 2 2 6 の D M A C 1 2 3 8 は書き込みコマンドを受け取り、このコマンドを B E 1 2 2 6 のスイッチ・ユニット 1 2 4 0 へ転送する。スイッチ・ユニット 1 2 4 0 は書き込みコマンドの中に含まれる D R A M アドレスを識別し、B E 1 2 2 6 のバンク・コントロール 1 2 4 2 を介して、D R A M 1 2 2 8 のバンク 1 2 4 4 へ、D R A M アドレス内に格納するデータを送る。したがって、スイッチ・ユニット 1 2 4 0 により、D R A M 1 2 2 4 と D R A M 1 2 2 8 の双方は、B E 1 2 2 2 の A P U 用の単一メモリ空間として機能することが可能になる。

30

【0042】

図 1 3 に D R A M の 6 4 個のバンク構成を示す。これらのバンクは 8 つの行 (1 3 0 2, 1 3 0 4, 1 3 0 6, 1 3 0 8, 1 3 1 0, 1 3 1 2, 1 3 1 4, 1 3 1 6) と 8 つの列 (1 3 2 0, 1 3 2 2, 1 3 2 4, 1 3 2 6, 1 3 2 8, 1 3 3 0, 1 3 3 2, 1 3 3 4) で構成されている。各行はバンク・コントローラにより制御される。したがって、各バンク・コントローラは 8 メガバイトのメモリを制御する。

【0043】

50

図14Aと14Bに最小のアドレス指定可能な格納単位（1024ビットのブロックなど）でのDRAMの格納とアクセスを行うための異なる構成を示す。図14Aでは、DMAC1402は單一のバンク1404の中に8つの1024ビット・ブロック1406を格納する。一方で図14Bでは、DMAC1412によって、1024ビットを含むデータ・ブロックの読み出しと書き込みが行われるが、これらのブロックは2つのバンク（バンク1414とバンク1416）の間で分配される。したがって、これらのバンクの各々には16個のデータ・ブロックが含まれ、データの各ブロックには512ビットが含まれる。この分配によって、DRAMのアクセスをさらに高速なものに改善することが可能となり、ある種のアプリケーションの処理に役立つ。

【0044】

図15にPE内のDMAC1504のアーキテクチャを示す。この図に示されているように、各APUがDMAC1506の構造上のノード1504へ直接アクセスを行うように、DMAC1506を含む構造上のハードウェアはPEを通じて配設される。各ノードは、ノードが直接アクセスを行う対象のAPUによるメモリ・アクセスに適した論理処理を実行する。

【0045】

図16にDMACの他の実施形態、すなわち、非分散型アーキテクチャを示す。この場合、DMAC1606の構造上のハードウェアは集中型である。APU1602とPUI604は、ローカルPEバス1607を介してDMAC1606を用いて通信を行う。DMAC1606はクロスバス交換機を介して、バス1608と接続される。バス1608はDRAM1610と接続されている。

【0046】

上述のように1つのPEの複数のAPUの全ては、独立に、共用DRAM内のデータへのアクセスが可能である。その結果、第1のAPUがあるデータをそのローカル・ストレージで処理しているときに、第2のAPUがこれらのデータを要求する場合もある。その時点で共用DRAMから第2のAPUへデータが取出された場合、データの値を変化させる第1のAPUの進行中の処理に起因して、そのデータが無効になる場合がある。したがって、その時点で第2のプロセッサが共用DRAMからデータを受け取った場合、第2のプロセッサでエラー結果が生じる恐れがある。例えば、このようなデータとしては、グローバル変数用の具体的な値が上げられる。第1のプロセッサがその処理中にその値を変えた場合、第2のプロセッサはもう使用されていない値を受け取ることになる。したがって、共用DRAMの範囲内に、メモリ・ロケーションからの、及び、メモリ・ロケーションへのAPUによるデータの読み出しと書き込みを同期させる何らかの方式が必要となる。この方法では、別のAPUがそのローカル・ストレージで現在働きかけている対象データであって、したがって最新のものではないデータのメモリ・ロケーションからの読み出しと、最新のデータを格納するメモリ・ロケーションの中へのデータの書き込みと、を行わないようにする必要がある。

【0047】

これらの問題を解決するために、DRAMの各アドレス指定が可能なメモリ・ロケーションに対して、そのメモリ・ロケーションに格納されているデータに関する状態情報を格納するために、DRAMの中でメモリの追加セグメントの割り振りが行われる。この状態情報の中には、フル・エンブティ(F/E)ビットと、メモリ・ロケーションからデータを要求するAPUの識別子(APU_ID)と、要求されたデータを読み出す読み出しが先となるAPUのローカル・ストレージのアドレス(LSアドレス)とが含まれる。DRAMのアドレス指定可能なメモリ・ロケーションは任意のサイズとすることができます。ある好みの実施形態では、このサイズは1024ビットである。

【0048】

F/Eビットの1への設定は、関連するメモリ・ロケーションに格納されているデータが最新のものであることを示す。一方、F/Eビットの0への設定は、関連するメモリ・ロケーションに格納されているデータが最新のものではないことを示す。このビットが0

10

20

30

40

50

に設定されているときに A P U がデータを要求しても、A P U によってそのデータの即時読み出しが妨げられる。この場合、そのデータを要求している A P U を識別する A P U I D と、データが最新のものになっているとき、そのデータを読み出す読み出しとなるこの A P U ローカル・ストレージ内のメモリ・ロケーションを識別する L S アドレスとが、追加メモリ・セグメントの中へ入力される。

【0049】

また追加メモリ・セグメントは、A P U のローカル・ストレージ内の各メモリ・ロケーションに対して割り振られる。この追加メモリ・セグメントは“ビジー・ビット”と呼ばれる 1 ビットを格納する。ビジー・ビットは D R A M から検索される固有データの格納用として、関連する L S メモリ・ロケーションの予約を行なうために使用される。ローカル・ストレージの特定のメモリ・ロケーションに対してビジー・ビットが 1 に設定されている場合、これらの固有データの書き込み用としてのみ、A P U はこのメモリ・ロケーションを使用することができる。一方、ローカル・ストレージの特定のメモリ・ロケーションに対して、ビジー・ビットが 0 に設定されている場合、A P U は任意のデータの書き込み用にこのメモリ・ロケーションを使用することができる。

【0050】

F/E ビット、A P U I D、L S アドレス、及びビジー・ビットが、P E の共用 D R A M からの、及び、P E の共用 D R A M へのデータの読み出しと書き込みを同期させるために使用される方法の例を、図 17 A から図 17 O に例示する。

【0051】

図 17 A に示すように、1 つ以上の P E (P E 1 7 2 0 など) が D R A M 1 7 0 2 を使用する。P E 1 7 2 0 には A P U 1 7 2 2 と A P U 1 7 4 0 とが含まれる。A P U 1 7 2 2 には制御論理回路 1 7 2 4 が含まれ、A P U 1 7 4 0 には制御論理回路 1 7 4 2 が含まれる。A P U 1 7 2 2 にはローカル・ストレージ 1 7 2 6 も含まれる。このローカル・ストレージには複数のアドレス可能なメモリ・ロケーション 1 7 2 8 が含まれる。A P U 1 7 4 0 にはローカル・ストレージ 1 7 4 4 が含まれ、このローカル・ストレージにも複数のアドレス可能なメモリ・ロケーション 1 7 4 6 が含まれる。これらのアドレス可能なメモリ・ロケーションのすべては、好適にはサイズが 1 0 2 4 ビットであることが望ましい。

【0052】

メモリの追加セグメントは各 L S のアドレス可能なメモリ・ロケーションと関連付けられる。例えば、メモリ・セグメント 1 7 2 9 と 1 7 3 4 とはそれぞれ、ローカルなメモリ・ロケーション 1 7 3 1 と 1 7 3 2 と関連付けられ、メモリ・セグメント 1 7 5 2 はローカル・メモリ・ロケーション 1 7 5 0 と関連付けられる。上述のような“ビジー・ビット”はこれらの追加のメモリ・セグメントの各々の中に格納される。ローカル・メモリ・ロケーション 1 7 3 2 は、このメモリ・ロケーションがデータを含むことを示すいくつかの X 印を用いて示されている。

【0053】

D R A M 1 7 0 2 には、メモリ・ロケーション 1 7 0 6 と 1 7 0 8 を含む、複数のアドレス可能なメモリ・ロケーション 1 7 0 4 を含まれる。またこれらのメモリ・ロケーションは、好適にはサイズが 1 0 2 4 ビットであることが望ましい。メモリの追加セグメントはまた、これらのメモリ・ロケーションの各々とも関連付けられる。例えば、追加のメモリ・セグメント 1 7 6 0 はメモリ・ロケーション 1 7 0 6 と関連付けられ、また、追加のメモリ・セグメント 1 7 6 2 はメモリ・ロケーション 1 7 0 8 と関連付けられる。各メモリ・ロケーションに格納されるデータに関連する状態情報は、メモリ・ロケーションに関連付けられたメモリに格納される。この状態情報の中には、上のように、F/E ビット、A P U I D、及び L S アドレスが含まれる。例えば、メモリ・ロケーション 1 7 0 8 については、この状態情報には、F/E ビット 1 7 1 2 、A P U I D 1 7 1 4 、及び L S アドレス 1 7 1 6 が含まれる。

【0054】

10

20

30

40

50

この状態情報とビシー・ピットとを用いて、P EのA P U、又は1グループのP E間での、共用D R A Mからの、および、同期した共用D R A Mからの読み出しと、同期した共用D R A Mへのデータの書き込みを行うことができる。

【0055】

図17Bに、A P U 1 7 2 2のL Sメモリ・ロケーション1 7 3 2から、D R A M 1 7 0 2のメモリ・ロケーション1 7 0 8へのデータの同期書き込みの開始を例示する図を示す。A P U 1 7 2 2の制御論理回路1 7 2 4によって、これらのデータの同期書き込みが開始される。メモリ・ロケーション1 7 0 8はエンブティであるため、F／Eピット1 7 1 2は0に設定される。その結果、メモリ・ロケーション1 7 0 8の中へL Sロケーション1 7 3 2内のデータを書き込むことが可能となる。このピットが1に設定され、メモリ・ロケーション1 7 0 8がフル状態であり、最新の有効データを含むことが示されている場合、制御回路1 7 2 4はエラー・メッセージを受け取ることになり、このメモリ・ロケーションへのデータの書き込みは禁止される。

【0056】

メモリ・ロケーション1 7 0 8への成功したデータの同期書き込みの結果を図17Cに示す。この書き込まれたデータはメモリ・ロケーション1 7 0 8に格納され、F／Eピット1 7 1 2は1に設定される。この設定により、メモリ・ロケーション1 7 0 8がフル状態であること、及び、このメモリ・ロケーションの中のデータが最新であり有効であることが示される。

【0057】

図17Dに、D R A M 1 7 0 2のメモリ・ロケーション1 7 0 8から、ローカル・ストレージ1 7 4 4のL Sメモリ・ロケーション1 7 5 0へのデータの同期読み出しの開始を例示する図を示す。この読み出しを開始するために、L Sメモリ・ロケーション1 7 5 0のメモリ・セグメント1 7 5 2の中のビジー・ピットが1に設定されて、このデータに対するメモリ・ロケーションが予約される。このビジー・ピットを1に設定することにより、A P U 1 7 4 0がこのメモリ・ロケーションにその他のデータを格納することはなくなっている。

【0058】

図17Eに示すように、制御論理回路1 7 4 2は次に、D R A M 1 7 0 2のメモリ・ロケーション1 7 0 8に対し、同期読み取りコマンドを出す。このメモリ・ロケーションに関連するF／Eピット1 7 1 2は1に設定されているので、メモリ・ロケーション1 7 0 8に格納されるデータは最新の有効データであると見なされる。その結果、メモリ・ロケーション1 7 0 8からL Sメモリ・ロケーション1 7 5 0へのデータ転送の準備の際に、F／Eピット1 7 1 2は0に設定される。この設定は図17Fに示されている。このピットを0に設定することは、これらのデータの読み出しの後に、メモリ・ロケーション1 7 0 8のデータは無効になることを示す。

【0059】

図17Gに示すように、メモリ・ロケーション1 7 0 8内のデータは次に、メモリ・ロケーション1 7 0 8からL Sメモリ・ロケーション1 7 5 0へ読み出される。図17Hは最終状態を示す。メモリ・ロケーション1 7 0 8のデータのコピーはL Sメモリ・ロケーションに格納される。F／Eピット1 7 1 2は0に設定され、メモリ・ロケーション1 7 0 8のデータは無効であることが示される。この無効はA P U 1 7 4 0によって行われた上記データの変更の結果である。メモリ・セグメント1 7 5 2のビジー・ピットも0に設定される。この設定によって、A P U 1 7 4 0がL Sメモリ・ロケーション1 7 5 0を任意の目的に利用できること、すなわち、このL Sメモリ・ロケーションがもはや固有データの受信を待機している予約状態ではないことが示される。したがって、L Sメモリ・ロケーション1 7 5 0は任意の目的のために、A P U 1 7 4 0によるL Sメモリ・ロケーション1 7 5 0へのアクセスが可能になる。

【0060】

図17Iから1 7 0に、D R A M 1 7 0 2のメモリ・ロケーション用F／Eピットが0

10

20

30

40

40

40

50

50

に設定され、このメモリ・ロケーションのデータが最新のものでもなく有効なものでもないことが示されている場合の、DRAM 1702 (メモリ・ロケーション 1708など) のメモリ・ロケーションから、APU のローカル・ストレージ (ローカル・ストレージ 1744 の LS メモリ・ロケーション 1752など) の LS メモリ・ロケーションへのデータの同期読み出しが例示されている。図 17I に示すように、この転送を開始するために、LS メモリ・ロケーション 1750 のメモリ・セグメント 1752 内のビジー・ビットは 1 に設定され、このデータ転送用としてこの LS メモリ・ロケーションが予約される。図 17J に示すように、制御論理回路 1742 は次に、DRAM 1702 のメモリ・ロケーション 1708 に対し、同期読み取りコマンドを出す。このメモリ・ロケーションと関連付けられた F/E ビット (F/E ビット 1712) は 0 に設定されているので、メモリ・ロケーション 1708 に格納されているデータは無効である。その結果、信号は制御論理回路 1742 へ伝送され、このメモリ・ロケーションからのデータの即時読み出しが阻止される。

【0061】

図 17K に示すように、APU ID 1714 とこの読み取りコマンド用の LS アドレス 1716 は次にメモリ・セグメントの中へ書き込まれる。この場合、APU 1740 用の APU ID と LS メモリ・ロケーション 1750 用の LS メモリロケーションは、メモリ・セグメント 1762 の中へ書き込まれる。したがって、メモリ・ロケーション 1708 のデータが最新のものになっているとき、この APU ID と LS メモリ・ロケーションは、最新のデータを伝送する伝送先のメモリ・ロケーションを決定するために使用される。

【0062】

メモリ・ロケーション 1708 のデータは、APU がこのメモリ・ロケーションの中へデータを書き込むと、有効で最新のデータとなる。APU 1722 のメモリ・ロケーション 1732 などから、メモリ・ロケーション 1708 への、データの同期書き込みが図 17L に示されている。このメモリ・ロケーション用の F/E ビット 1712 が 0 に設定されているため、これらのデータのこの同期書き込みは許される。

【0063】

図 17M に示すように、この書き込み後、メモリ・ロケーション 1708 のデータは最新の有効データになる。したがって、メモリ・セグメント 1762 から得られる APU ID 1714 と LS アドレス 1716 とは、メモリ・セグメント 1762 から即座に読み出され、次いで、この情報はこのセグメントから削除される。メモリ・ロケーション 1708 のデータの即時読み出しを予測して、F/E ビット 1712 もまた 0 に設定される。図 17N に示すように、APU ID 1714 と LS アドレス 1716 を読み出すと、APU 1740 の LS メモリ・ロケーション 1750 へメモリ・ロケーション 1708 の有効データを読み出すために、この情報は直ちに使用される。図 17O に最終状態を示す。この図は、メモリ・ロケーション 1708 からメモリ・ロケーション 1750 にコピーされた有効データと、0 に設定されたメモリ・セグメント 1752 のビジー・ビットと、0 に設定されたメモリ・セグメント 1762 の F/E ビット 1712 を示す。このビジー・ビットを 0 に設定することにより、任意の目的のために APU 1740 が LS メモリ・ロケーション 1750 のアクセスを行うことができる。この F/E ビットを 0 に設定することにより、メモリ・ロケーション 1708 のデータがもはや最新のものではなく、有効なものでもないことが示される。

【0064】

図 18 は上述のオペレーションと、DRAM のメモリ・ロケーションの様々な状態とを要約する図であり、この状態は F/E ビットの状態と、APU ID と、メモリ・ロケーションに対応するメモリ・セグメントの中に格納された LS アドレスとに基づく。このメモリ・ロケーションは、3 つの状態を持つことが可能である。これらの 3 つの状態として、F/E ビットが 0 に設定され、APU ID または LS アドレスに対して情報が提供されないエンブティ状態 1880 と、F/E ビットが 1 に設定され、APU ID または L

Sアドレスに対して情報が提供されないフル状態1882と、F/Eビットが0に設定され、APU-IDとLSアドレスに対して情報が提供されるブロッキング状態1884とがある。

【0065】

この図に示すように、エンブティ状態1880では、同期書き込みオペレーションが許され、フル状態1882への遷移という結果が得られる。しかし、メモリ・ロケーションがエンブティ状態であるときは、メモリ・ロケーション内のデータが最新のものではないので、同期読み出しオペレーションに対しては、ブロッキング状態へ遷移するという結果となる。

【0066】

フル状態1882では、同期読み出しオペレーションが許され、エンブティ状態1880への遷移という結果が得られる。一方、有効データの上書きを避けるために、フル状態1882の同期書き込みオペレーションは禁止される。このような書き込みオペレーションがこの状態で試みられる場合、状態の変化は生じず、エラー・メッセージがAPUの対応する制御論理回路へ伝送される。

【0067】

ブロッキング状態1884では、メモリ・ロケーションの中へのデータの同期書き込みが許され、エンブティ状態1880への遷移という結果が得られる。一方、ブロッキング状態1884での同期読み出しオペレーションは禁止される。それは、このブロッキング状態を生じさせることになった前回の同期読み出しオペレーションとのコンフリクトを阻止するためである。同期読み出しオペレーションがブロッキング状態1884で試みられた場合、状態変化は生じず、APUの対応する論理制御回路へエラー・メッセージが伝送される。

【0068】

共用DRAMからのデータの同期読み出しと、共用DRAMへのデータの同期書き込みを行う上述の方式は、外部デバイスからのデータ読み出しと、外部デバイスのデータ書き込み用プロセッサとして通常専用の計算用リソースを省くためにも利用することができる。この入力/出力(I/O)機能はPUによって行うこともできる。しかし、この同期方式の変更を利用して、適切なプログラムを実行するAPUがこの機能を実行してもよい。例えば、この方式を利用して、外部デバイスによって開始されたI/Oインターフェースからのデータ伝送を求める割込み要求を受け取るPUは、このAPUにこの要求処理を委任してもよい。次にAPUは、I/Oインターフェースに対して同期書き込みコマンドを出す。今度はこのインターフェースによって、現在DRAMの中へデータを書き込むことができる旨の信号が外部デバイスへ送られる。次にAPUはDRAMへ同期読み取りコマンドを出し、DRAMの関連するメモリ空間をブロッキング状態に設定する。APUはまた、データを受け取る必要があるAPUのローカル・ストレージのメモリ・ロケーションに対して、ビジー・ビットを1に設定する。ブロッキング状態では、DRAMの関連するメモリ空間と関連付けられた追加メモリ・セグメントの中に、APUのIDとAPUのローカル・ストレージの関連するメモリ・ロケーションのアドレスが含まれる。次に外部デバイスは同期書き込みコマンドを出し、DRAMの関連するメモリ空間へデータが直接書き込まれる。このメモリ空間はブロッキング状態にあるので、データは、この空間の中から、追加メモリ・セグメントの中で識別されたAPUのローカル・ストレージのメモリ・ロケーションの中へ直ちに読み出される。次いで、これらのメモリ・ロケーション用のビジー・ビットは0に設定される。外部デバイスがデータの書き込みを完了したとき、APUは伝送が完了した旨を示す信号をPUへ出す。

【0069】

したがって、この方式を用いて、PUに対する最小の計算上の負荷で、外部デバイスからのデータ転送処理を行うことができる。しかし、この機能を任せられたAPUはPUに対して割込み要求を出せることが望ましく、外部デバイスがDRAMに対して直接アクセスを行うことが望ましい。

10

20

30

40

50

【0070】

各 P E の D R A M には複数の“サンドボックス”が含まれる。サンドボックスによって共用 D R A M 領域が拘束され、この領域を超えて、特定の A P U または 1 組の A P U がデータの読み出しや書き込みを行うことはできない。これらのサンドボックスによって、1つの A P U が処理するデータに起因する、別の A P U によって処理されるデータの破損に対するセキュリティが与えられる。またこれらのサンドボックスによって、ソフトウェア・セルが全 D R A M の中でデータの破損を生じる可能性はなく、ネットワーク 1 0 4 から特定のサンドボックスの中へソフトウェア・セルのダウンロードを行なうことが許される。本発明では、サンドボックスは D R A M と D M A C とからなるハードウェアの中に設けられる。ソフトウェアの代わりに、このハードウェア内にこれらのサンドボックスを設けることにより、速度とセキュリティという利点が得られる。

10

【0071】

P E の P U は A P U へ割り当てられるサンドボックスの制御を行う。P U は通常、オペレーティング・システムのような信頼のにおけるプログラムだけしか動作させないので、この方式によって、セキュリティが危険にさらされることはない。この方式に従って、P U はキー管理テーブルの構築と維持を行なう。図 1 9 にこのキー管理テーブルを示す。この図に示すように、キー管理テーブル 1 9 0 2 の各エントリには、A P U 用の識別子 (I D) 1 9 0 4 と、その A P U 用の A P U キー 1 9 0 6 と、キー・マスク 1 9 0 8 とが含まれる。このキー・マスクの用途について以下に説明する。キー管理テーブル 1 9 0 2 は好適にはスタティック・ランダム・アクセス・メモリ (S R A M) のような比較的高速のメモリに格納され、D M A C と関連付けられる。キー管理テーブル 1 9 0 2 へのエントリは P U によって制御される。A P U が D R A M の特定格納位置 (ストレージロケーション) へのデータの書き込みとあるいは D R A M の特定の格納位置からのデータの読み出しを要求すると、D M A C はその格納位置と関連付けられたメモリ・アクセス・キーに対してキー管理テーブル 1 9 0 2 内のその A P U へ割り当てられた A P U キー 1 9 0 6 の評価を行なう。

20

【0072】

図 2 0 に示すように、D R A M 2 0 0 2 の各アドレス可能な格納位置 2 0 0 6 に対して専用メモリ・セグメント 2 0 1 0 が割り当てられる。この格納位用のメモリ・アクセス・キー 2 0 1 2 はこの専用メモリ・セグメントの中に格納される。上述のように、やはり各アドレス可能格納位置 2 0 0 6 と関連付けられたさらなる追加専用メモリ・セグメント 2 0 0 8 によって、格納位置へのデータを書き込みと、格納位置からのデータ読み出しを行なうための同期情報が格納される。

30

【0073】

作動時に、A P U は D M A C へ D M A コマンドを出す。このコマンドには D R A M 2 0 0 2 の格納位置 2 0 0 6 のアドレスが含まれる。このコマンドを実行する前に、D M A C は、キー管理テーブル 1 9 0 2 における A P U の I D 1 9 0 4 を用いて要求を行なっている A P U のキー 1 9 0 6 を調べる。次いで D M A C は A P U がアクセスを求める対象先である D R A M の格納位置と関連付けられた専用メモリ・セグメント 2 0 1 0 の中に格納されるメモリ・アクセス・キー 2 0 1 2 と、要求を行なっている A P U の A P U キー 1 9 0 6 との比較を行う。2 つのキーが一致しない場合、D M A コマンドは実行されない。一方、2 つのキーが一致する場合、D M A コマンドは進行し、要求されたメモリ・アクセスが実行される。

40

【0074】

図 2 1 に他の実施形態を例示する。この実施形態では、P U はメモリ・アクセス管理テーブル 2 1 0 2 の維持も行う。メモリ・アクセス管理テーブル 2 1 0 2 には D R A M 内にある各サンドボックス用のエントリが含まれる。図 2 1 の特定の例では、D R A M には 6 個のサンドボックスが含まれる。メモリ・アクセス管理テーブル 2 1 0 2 内の各エントリには、サンドボックス用識別子 (I D) 2 1 0 4 と、ベース・メモリ・アドレス 2 1 0 6 と、サンドボックス・サイズ 2 1 0 8 と、メモリ・アクセス・キー 2 1 1 0 と、アクセ

50

ス・キー・マスク 2112 が含まれる。ベース・メモリ・アドレス 2106 によって、DRAM にアドレスが設定されたアドレスによって特定のメモリ・サンドボックスの最初の部分が示される。サンドボックス・サイズ 2108 によりサンドボックスのサイズが与えられ、したがって、このサイズにより特定のサンドボックスのエンドポイントが与えられる。

【0075】

図 22 はキー管理テーブル 1902 とメモリ・アクセス管理テーブル 2102 を用いて DMA コマンドを実行するためのステップを示すフローチャートである。ステップ 2202 では、APU によってサンドボックス内の特定 1つあるいは複数のメモリ・ロケーションに対するアクセス用 DMA コマンドが DMAC へ出される。このコマンドにはアクセス要求を行う対象先である特定のサンドボックスの識別を行なうサンドボックス ID 2104 が含まれる。ステップ 2204 で、DMAC は APU の ID 1904 を利用して、キー管理テーブル 1902 内の要求を行なっている APU のキー 1906 を調べる。ステップ 2206 で、DMAC はメモリ・アクセス管理テーブル 2102 で、サンドボックスと関連付けられたメモリ・アクセス・キー 2110 を調べるコマンドで、サンドボックス ID 2104 を利用する。ステップ 2208 で、DMAC は、要求を行なっている APU へ割り当てられている APU キー 1906 と、サンドボックスと関連付けられたアクセス・キー 2110 と比較する。ステップ 2210 で、2つのキーが一致するかどうかの決定が行われる。2つのキーが一致しない場合、処理はステップ 2212 へ移行し、そこで DMA コマンドは先へ進まず、要求を行なっている APU と PU のいずれか、または双方へエラーメッセージが送信される。一方、ステップ 2210 で、2つのキーの一一致が得られた場合、処理はステップ 2214 へと進み、そこで DMAC は DMA コマンドを実行する。

10

20

【0076】

APU 用キーとメモリ・アクセス・キー用のキー・マスクにより、このシステムに大きな柔軟性が与えられる。キー用のキー・マスクにより、マスクされたビットはワイルド・カードに変換される。例えば、APU キー 1906 と関連付けられたキー・マスク 1908 が、キー・マスク 1908 内のこれらのビットを 1 に設定することなどにより、その最後の 2 ビットが“マスク”に設定されている場合、APU キーは 1 又は 0 のいずれかになることができ、そのままメモリ・アクセス・キーに一致することになる。例えば、APU キーは 1010 であるとする。通常、この APU キーによって 1010 のアクセス・キーを持つサンドボックスへのアクセスだけが可能になる。しかし、この APU キー用の APU キー・マスクが 0001 に設定されている場合、この APU キーを用いて 1010 または 1011 のいずれかのアクセス・キーを持つサンドボックスへのアクセスを行うことが可能となる。同様に、1010 または 1011 のいずれかの APU キーを持つ APU によって、0001 に設定されたマスクを持つアクセス・キー 1010 のアクセスを行うことが可能である。APU キー・マスクとメモリ・キーマスクの双方を同時に使用することができるでの、多数のバリエーションのサンドボックスに対する APU によるアクセシビティの設定が可能となる。

30

【0077】

本発明はまた、システム 101 のプロセッサ用の新しいプログラミング・モデルも提供する。このプログラミング・モデルではソフトウェア・セル 102 が用いられる。ネットワーク 104 上の任意のプロセッサへ処理用として、これらのセルの伝送を行うことが可能である。また、この新しいプログラミング・モデルでは、システム 101 のユニークなモジュラー形アーキテクチャとシステム 101 のプロセッサとが利用される。

40

【0078】

ソフトウェア・セルは APU のローカル・ストレージから APU によって直接処理される。APU は DRAM 内のいずれのデータ又はプログラムに対して直接書きかけることは行わない。DRAM 内のデータとプログラムは、APU がこれらのデータとプログラムの処理を行う前に、APU のローカル・ストレージの中に読み込まれる。したがって、APU のローカル・ストレージには、プログラム・カウンタと、これらのプロ

50

グラムを実行するための他のソフトウェア・エレメントとが含まれる。P UはDMA Cに対してDMAコマンドを出すことによりA P Uの制御を行う。

【0079】

図23にソフトウェア・セル102の構造を例示する。この図に示すように、ソフトウェア・セル2302などのソフトウェア・セルの中には、ルート選定情報セクション2304と本体部分2306などが含まれる。ルート選定情報セクション2304に含まれる情報は、ネットワーク104のプロトコルに依って決められる。ルート選定情報セクション2304にはヘッダ2308、宛先ID2310、ソースID2312、及び応答ID2314が含まれる。宛先ID2310にはネットワーク・アドレスが含まれる。TCP/IPプロトコルの下で、例えば、ネットワーク・アドレスはインターネット・プロトコル(I/P)である。更に、宛先ID2310には、処理のためにセルを伝送すべき伝送先のP E及びA P Uの識別子が含まれる。ソースID2314にはネットワーク・アドレスが含まれ、このソースIDによってP EとA P Uとが識別され、このP EとA P Uとからセルが起動し、必要な場合に、宛先P EとA P Uとがセルに関する追加情報を得ることが可能となる。応答ID2314にはネットワーク・アドレスが含まれ、この応答ID2314によって、セルに関するクエリとセルの処理結果とを送る送り先のP EとA P Uとが識別される。

【0080】

セルの本体部分2306にはネットワークのプロトコルとは無関係の情報が含まれる。図23の分解部分は、セルの本体部分2306の細部を示す。セルの本体部分2306のヘッダ2320によってセル本体部分の開始部が識別される。セル・インターフェース2322にはセルの利用に必要な情報が含まれる。この情報にはグローバルな一意のID2324と要求されるA P U2326と、サンドボックス・サイズ2328と、前回のセルのID2330が含まれる。

【0081】

グローバルな一意的ID2324はネットワーク104全体を通じてソフトウェア・セル2302を一意的に識別する。グローバルな一意的ID2324はソースID2312(ソースID2312内のP E又はA P Uの一意的識別子など)と、ソフトウェア・セル2302の作成又は伝送の瞬刻と日付に基づき作成される。必要なA P U2326によってセルの実行に必要な最低数のA P Uが与えられる。サンドボックス・サイズ2328により、セルの実行に必要なD R A Mと関連する必要なA P U内に、保護されたメモリ量が与えられる。前回のセルID2330により、シーケンシャルな実行を要求する1グループのセル(ストリーミング・データなど)内の前回のセルの識別子が提供される。

【0082】

実行セクション2332にはセルのコア情報が含まれる。この情報には、DMAコマンド・リスト2334と、プログラム2336と、データ2338とが含まれる。プログラム2336には、A P Uプログラム2360や2338などの、A P Uによって実行されるプログラム(“アブレット”と呼ばれる)が含まれ、データ2338にはこれらのプログラムを用いて処理されるデータが含まれる。DMAコマンド・リスト2334には、プログラムの起動に必要な一連のDMAコマンドが含まれる。これらのDMAコマンドには、DMAコマンド2340、2350、2355、2358が含まれる。P UはDMA CへこれらのDMAコマンドを出す。

【0083】

DMAコマンド2340にはVID2342が含まれる。VID2342は、DMAコマンドが出されたときに、物理IDに対して対応付けられるA P UのバーチャルIDである。DMAコマンド2340には、ロード・コマンド2344とアドレス2346も含まれる。ロード・コマンド2344はA P UにD R A Mから特定の情報を読み出して、ロカル・ストレージの中へ入れるように命令する。アドレス2346によって、この特定情報を含むD R A M内のバーチャル・アドレスが与えられる。この特定情報は、プログラム・セクション2336からのプログラム、データ・セクション2338からのデータ、又

10

20

30

40

50

はその他のデータなどであってよい。最終的に、DMAコマンド2340にはローカル・ストレージのアドレス2348が含まれる。このアドレスにより、情報をロードできそうなローカル・ストレージのアドレスが識別される。DMAコマンド2350には同様の情報が含まれる。その他のコマンドも使用可能である。

【0084】

DMAコマンド・リスト2334には、一連のキック・コマンド（キック・コマンド2355や2358など）も含まれる。キック・コマンドとは、PUによってAPUへ出される、セルの処理を開始するコマンドである。DMAキック・コマンド2355にはバーチャル・APU ID2352と、キック・コマンド2354と、プログラム・カウンタ2356とが含まれる。バーチャル・APU ID2352はキックすべき対象APUを識別し、キック・コマンド2354は関連するキック・コマンドを与え、プログラム・カウンタ2356はプログラムの実行用プログラム・カウンタのためのアドレスを与える。DMAキック・コマンド2358は同じAPU又は別のAPUに対して同様の情報を与える。

【0085】

上述のように、PUは独立したプロセッサとしてAPUを扱い、コプロセッサとして扱うものではない。したがって、APUによる処理を制御するために、PUは遠隔手順呼出しに類似したコマンドを使用する。これらのコマンドは“APU遠隔手順呼出し（ARPC）”と呼ばれる。PUは一連のDMAコマンドをDMACへ出すことにより、ARPCを実行する。DMACはAPUプログラムとそれに関連するスタック・フレームなどをAPUのローカル・ストレージへロードする。次いでPUはAPUへ最初のキックを出し、APUプログラムを実行する。

10

【0086】

図24にアプレットを実行するためのARPCのステップを例示する。指定のAPUによるアプレットの処理の開始時にPUが実行するこれらのステップが、図24の第1の部分2404に示されている。

20

【0087】

ステップ2410で、PUはアプレットを評価し、次にアプレットの処理用APUを指定する。ステップ2412で、PUは、必要な単複のサンドボックス用のメモリ・アクセス・キーの設定を行うDMAコマンドをDMACへ出すことにより、アプレットの実行用スペースをDRAM内に割り振る。ステップ2414で、PUは指定APUへの割込み要求による、アプレットの完了信号の伝送を可能にする。ステップ2418で、PUはDRAMからAPUのローカル・ストレージへアプレットをロードするDMAコマンドをDMACへ出す。ステップ2420で、DMAコマンドが実行され、DRAMからAPUのローカル・ストレージへアプレットが読み出される。ステップ2422で、PUは、アプレットと関連付けられたスタック・フレームをDRAMからAPUのローカル・ストレージへロードするDMAコマンドをDMACへ出す。ステップ2423で、DMAコマンドが実行され、スタック・フレームがDRAMからAPUのローカル・ストレージへ読み出される。ステップ2424で、PUは、DMACがAPUへキーを割り当てて、ステップ2412で指定された、一又は複数のハードウェア・サンドボックスからのデータ読み出しと、その一又は複数のハードウェア・サンドボックスへのデータ書き込みを行なうことをAPUに許可するDMAコマンドを出す。ステップ2426で、DMACはAPUへ割り当てられたキーを用いて、キー管理テーブル（KTAB）の更新を行う。ステップ2428で、PUは、プログラムの処理を開始するDMAコマンド“キック”をAPUに出す。特定のアプレットに応じて、特定のARPCの実行時に、PUによって他のDMAコマンドを出しててもよい。

30

【0088】

上述のように、図24の第2の部分2404はアプレットの実行時にAPUにより行われるステップを例示するものである。ステップ2430で、APUは、ステップ2428で出されるキック・コマンドに応じてアプレットの実行を開始する。ステップ2432で

40

50

、アブレットの指示で、APUはアブレットの関連スタック・フレームの評価を行う。ステップ2434で、APUはDMACへ複数のDMAコマンドを出し、スタック・フレームが必要に応じてDRAMからAPUのローカル・ストレージへ指定するデータのロードを行う。ステップ2436で、これらのDMAコマンドが実行され、データはDRAMからAPUのローカル・ストレージへ読み出される。ステップ2438で、APUはアブレットを実行し、ある結果を出力する。ステップ2440で、APUはDMACへDMAコマンドを出し、DRAMにその結果を格納する。ステップ2442で、DMAコマンドが実行され、アブレットの結果がAPUのローカル・ストレージからDRAMへ書き込まれる。ステップ2444で、APUはPUへ割込み要求を出し、ARPCが完了したことを示す信号伝送を行う。

10

【0089】

PUの指示の下で独立にタスクを実行するAPUの能力によって、1グループのAPUと、1グループのAPUと関連付けられたメモリ・リソースとを拡張タスクの実行専用にすることが可能になる。例えば、1つのPUは、1以上のAPUと、これらの1以上のAPUと関連付けられた1グループのメモリ・サンドボックスとを、拡張された時間中、ネットワークを介して伝送されてくるデータの受信専用とし、また、1以上の他のAPUとそれらと関連付けられたメモリ・サンドボックスへ、この時間中受信したデータの異なる処理を行うための送信専用とすることができます。この能力は、ネットワーク104を介して伝送されるストリーミングデータ（ストリーミングMPPEG又はストリーミングATRACオーディオ又はビデオ・データなど）の処理によって特に好適である。PUは、1以上のAPU及びそれらと関連付けられたメモリ・サンドボックスをこれらのデータの受信専用とし、1以上の他のAPU及びそれらと関連付けられたメモリ・サンドボックスをこれらのデータの解凍と処理専用とすることができます。言い換えれば、PUはAPUのグループとそれらと関連付けられたメモリ・サンドボックスとの間でこのようなデータ処理を行いうための専用パイプライン関係の確立を行なうことができる。

20

【0090】

しかし、このような処理を効率的に実行するためには、パイプラインの専用APUとメモリ・サンドボックスとが、データ・ストリームを含むアブレットの処理が行われない時間中もパイプライン専用のままであることが望ましい。言い換えれば、専用APUとその関連付けられたサンドボックスが、これらの時間中、予約状態のままに置かれることが望ましい。アブレットの処理の完了時ににおける、APUとその関連付けられた一又は複数のメモリ・サンドボックスを予約、すなわちリザーブ状態としておくことは、“常駐終了”と呼ばれる。常駐終了はPUからの命令に応じて行われる。

30

【0091】

図25、26A及び26Bは、1グループのAPU及びそれらと関連するサンドボックスを含む、ストリーミングデータ（ストリーミングMPPEGデータなど）を処理するための専用パイプライン構造の設定を例示する。図25に示すように、このパイプライン構造の構成要素にはPE2502とDRAM2518とが含まれる。PE2502にはPU2504とDMAC2506とが含まれ、複数のAPUには、APU2508と、APU2510と、APU2512とが含まれる。広帯域バス2516によりDMAC2506はDRAM2518と接続される。DRAM2518には複数のサンドボックス（サンドボックス2520、サンドボックス2522、サンドボックス2524、サンドボックス2526など）が含まれる。

40

【0092】

図26Aに専用パイプラインを設定するためのステップを例示する。ステップ2610で、PU2504はネットワーク・アブレットを処理するようにAPU2508を割り当てる。ネットワーク・アブレットはネットワーク104のネットワーク・プロトコルの処理用プログラムを有する。この場合、このプロトコルは、伝送制御プロトコル/インターネット用プロトコル（TCP/IP）である。このプロトコルに従うTCP/IPデータ・パケットは、ネットワーク104を介して伝送される。受信時に、APU2508はこ

50

これらのパケットを処理し、パケット内のデータを組み立て、ソフトウェア・セル102の中へ入れる。ステップ2612で、P U 2 5 0 4は、ネットワーク・アプレットの処理の完了時に常駐終了を実行するようにA P U 2 5 0 8に指示する。ステップ2614で、P U 2 5 0 4は、A P U 2 5 1 0及び2512がM P E Gアプレットの処理を行うように割り当てる。ステップ2615で、P U 2 5 0 4はM P E Gアプレットの処理の完了時に常駐終了を実行するようにA P U 2 5 1 0と2512に指示する。ステップ2616で、P U 2 5 0 4は、A P U 2 5 0 8とA P U 2 5 1 0によるアクセス用ソース・サンドボックスとしてサンドボックス2520を指定する。ステップ2618で、P U 2 5 0 4はA P U 2 5 1 0によるアクセス用优先サンドボックスとしてサンドボックス2522を指定する。ステップ2620で、P U 2 5 0 4は、A P U 2 5 0 8とA P U 2 5 1 2によるアクセス用ソース・サンドボックスとしてサンドボックス2524を指定する。ステップ2622で、P U 2 5 0 4は、A P U 2 5 1 2によるアクセス用优先サンドボックスとしてサンドボックス2526を指定する。ステップ2624で、A P U 2 5 1 0とA P U 2 5 1 2とは、それぞれ、ソース・サンドボックス2520とソース・サンドボックス2524の範囲内のメモリ・ブロックへ同期読み取りコマンドを送り、これらのメモリ・ブロックをプロッキング状態に設定する。最後に、処理はステップ2628へ移り、そこで専用のバイオペーラインの設定が完了し、バイオペーライン専用のリソースが予約される。このようにして、A P U 2 5 0 8、2510、2512及びそれらと関連するサンドボックス2520、2522、2524、及び2526は予約状態に入る。

【0093】

図26Bにこの専用バイオペーラインによるストリーミングM P E Gデータの処理ステップを示す。ステップ2630で、A P U 2 5 0 8はネットワーク・アプレットを処理し、そのローカル・ストレージの中でT C P / I Pデータパケットをネットワーク104から受信する。ステップ2632で、A P U 2 5 0 8はこれらのT C P / I Pデータ・パケットを処理し、これらのパケット内のデータを組み立て、ソフトウェア・セル102の中へ入れる。ステップ2634で、A P U 2 5 0 8はソフトウェア・セルのヘッダ2320(図23)をチェックし、セルがM P E Gデータを含むかどうかの判定を行う。セルがM P E Gデータを含まない場合、ステップ2636で、A P U 2 5 0 8は専用バイオペーラインに含まれない他のA P Uによって他のデータを処理するために、D R A M 2 5 1 8内に指定される汎用サンドボックスへセルを伝送する。A P U 2 5 0 8はこの伝送についてP U 2504に通知する。

【0094】

一方、ソフトウェア・セルがM P E Gデータを含む場合は、ステップ2638で、A P U 2 5 0 8はそのセルの前のセルI D 2 3 3 0(図23)をチェックし、そのセルが属するM P E Gデータ・ストリームを識別する。ステップ2640で、A P U 2 5 0 8はセル処理用の専用バイオペーラインのA P Uを選択する。この場合、A P U 2 5 0 8はこれらのデータを処理するA P U 2 5 1 0を選択する。この選択は前回のセルI D 2 3 3 0とロード・バランスシング・ファクタ(負荷平衡係数)とに基づく。例えば、そのソフトウェア・セルが属するM P E Gデータ・ストリームの前回のソフトウェア・セルが処理用としてA P U 2 5 1 0へ送られたことが前回のセルI D 2 3 3 0によって示されている場合、現在のソフトウェア・セルも通常の処理用としてA P U 2 5 1 0へ送られる。ステップ2642で、A P U 2 5 0 8は、サンドボックス2520へM P E Gデータを書き込む、同期書き込みコマンドを出す。このサンドボックスはあらかじめプロッキング状態に設定されているので、ステップ2644で、M P E Gデータはサンドボックス2520からA P U 2 5 1 0のローカル・ストレージへ自動的に読み出される。ステップ2646で、A P U 2 5 1 0はそのローカル・ストレージでM P E Gデータを処理してビデオ・データを生成する。ステップ2648で、A P U 2 5 1 0はサンドボックス2522へビデオ・データを書き込む。ステップ2650で、A P U 2 5 1 0は同期読み出しコマンドをサンドボックス2520へ出し、このサンドボックスに追加M P E Gデータ受信の準備をする。ステップ2652で、A P U 2 5 1 0は常駐完了処理を行う。この処理により、このA P Uは予約

10

20

30

40

50

状態に入り、この予約状態の間、A P U は M P E G データ・ストリームの中で追加 M P E G データの処理を行うべく待機する。

【0095】

他のタイプのデータ処理用として、1 グループの A P U 及びそれらと関連するサンドボックス間でその他の専用構造の設定が可能である。例えば、図 27 に示すように、A P U の専用グループ（A P U 2 7 0 2、2 7 0 8、2 7 1 4 など）を設定し、3 次元オブジェクトに対して幾何学変換を実行して 2 次元ディスプレイ・リストの生成を行なうことが可能となる。これらの 2 次元ディスプレイ・リストを他の A P U によって更に処理（レンダリング）、画素データの生成を行うようになることが可能である。この処理を実行するために、3 次元オブジェクトと、これらのオブジェクト処理から結果として生じるディスプレイ・リストの格納用として、サンドボックスが、A P U 2 7 0 2、2 7 0 8、2 4 1 4 の専用となる。例えば、ソース・サンドボックス 2 7 0 4、2 7 1 0、2 7 1 6 は、それぞれ、A P U 2 7 0 2、2 7 0 8、2 7 1 4 によって処理された 3 次元オブジェクトの格納専用となる。同様に、宛先サンドボックス 2 7 0 6、2 7 1 2、2 7 1 8 はそれぞれ、A P U 2 7 0 2、A P U 2 7 0 8、A P U 2 7 1 4 によるこれらの 3 次元オブジェクトの処理から結果として生じるディスプレイ・リストの格納専用となる。

【0096】

調整用 A P U 2 7 2 0 は、そのローカル・ストレージにおける、宛先サンドボックス 2 7 0 6、2 7 1 2、2 7 1 8 からのディスプレイ・リストの受信専用である。A P U 2 7 2 0 はこれらのディスプレイ・リスト間で調整を行い、画素データのレンダリングのためにこれらのディスプレイ・リストを他の A P U へ送る。

10

20

【0097】

システム 1 0 1 のプロセッサは絶対タイマーも使用する。この絶対タイマーは A P U と P E の他のエレメントへクロック信号を出力する。このクロック信号はこれらのエレメントを駆動するクロック信号に依存せず、かつ、このクロック信号より高速である。この絶対タイマーの利用が図 28 に例示されている。

30

【0098】

この図に示すように、絶対タイマーによって A P U によるタスク・パフォーマンスのためのタイム・バジェット（割り当て時間）が決定される。このタイム・バジェットによって、これらのタスクの完了時間が設定されるが、この時間は A P U によるタスク処理に必要な時間より長い時間になる。その結果、各タスクについて、タイム・バジェットの範囲内に、ビジーな時間とスタンバイ時間とが存在することになる。全てのアプレットは、A P U の実際の処理時間あるいは速度にかかわらず、このタイム・バジェットに基づいて処理を行うように書かれている。例えば、P E の特定の A P U 用として、タイム・バジェット 2 8 0 4 のビジー時間 2 8 0 2 中に特定のタスクを行なうことができる。ビジー時間 2 8 0 2 がタイム・バジェット 2 8 0 4 未満であるため、スタンバイ時間 2 8 0 6 はタイム・バジェット中に生じる。このスタンバイ時間中、A P U は A P U が消費するパワーが少なくなるスリープモードに入る。

【0099】

タイム・バジェット 2 8 0 4 が満了するまで、他の A P U 又は P E の他のエレメントがタスク処理の結果を予想することはしない。したがって、A P U の実際の処理速度にかかわらず、絶対タイマーによって決定されるタイム・バジェットを用いて、A P U の処理結果が常時調整される。

40

【0100】

将来、A P U による処理速度は更に高速になる。しかし、絶対タイマーによって設定されるタイム・バジェットは同じままである。例えば、図 28 に示すように、将来の A P U は更に短時間でタスクを実行することになり、したがって、スタンバイ時間は更に長くなるであろう。したがって、ビジー時間 2 8 0 8 はビジー時間 2 8 0 2 よりも短くなり、スタンバイ時間 2 8 1 0 はスタンバイ時間 2 8 0 6 よりも長くなる。しかし、絶対タイマーによって設定された同じタイム・バジェットに基づいて処理を行なうようにプログラムが書

50

かれているので、A P U間の処理結果の調整が維持される。その結果、更に高速のA P Uが、その処理の結果が予測される時点でコンフリクトを生じることなく、低速のA P U用として書かれたプログラムの処理を行うことが可能となる。

【0101】

動作速度の向上や動作速度が異なることに起因するA P Uの並列処理の調整問題に対しては、A P U間での調整を決定する絶対タイマーに代えて、P Uまたは1以上の指定A P Uにおいて、A P Uが実行している特定の命令（マイクロコード）の分析をアプレットの処理時に使うようにすることもできる。“オペレーションなし”（“N O O P”）命令を命令の中へ挿入し、A P Uのいくつかによってこの命令を実行してアプレットによって予測されるA P Uによる処理を1ステップずつ適切に行なうことが可能となる。命令の中へこれらのN O O Pを挿入することにより、全ての命令のA P Uによる実行を行なうための正しいタイミングの維持が可能となる。

10

【0102】

上述のように、各処理エレメント（P E）には少なくとも1つの処理ユニット（P U）と、1つ以上の複数の附加処理デバイス（A P U）が含まれ、P Uの調整及び制御下で、1つ以上のアプリケーションによりデータの並列処理がA P Uによって実行される。これに連動して、また、手短に図18を参照すると、図18にはA P Uと共にメモリとの間にデータをやりとりして送るための、例示的なデータレベル同期（D L S）機構の状態が示されていた。しかし、前述のように、このD L S機構は外部デバイスへの、及び／又は外部デバイスからのデータの処理にも好適に利用される。更に、以下の説明からわかるように、I／Oデバイスを扱うためにD L Sを利用することにより、その他の利点が与えられる。例えば、現在のところそれらの利点としては、デバイス実装の詳細を隠す能力、複数のデバイスを扱う柔軟性、1つ以上の外部ソースからのストリーミング・データの処理を行なうA P Uの能力、が挙げられる。図29に留意すると、図29には本発明の概念による例示的なD L Sアーキテクチャが示されている。

20

【0103】

図29において、構成2900にはP E 2900と、D R A M（又は共用メモリ）2910と、インターフェースデバイス（インターフェース）2915と、外部デバイス2920とが含まれる。P E 2990は1つ以上の処理エレメントを示し、さらにP E 2900には、少なくとも1つのP U 2905と、1つ以上のA P U（A P U 2901、2902、2903により示す）とが含まれる。P E 2990はバス2906を介して共用メモリと結合されており、またP E 2990は前述のD L S機構とメモリ保護機構を用いて、D R A M 2910へのデータの書き込みと、D R A M 2910からのデータの読み出しを行う。本発明の概念によると、現在のところ、このD L S機構は外部デバイスと用いるよう拡張されている。そのため、また図29からわかるように、P E 2990はバス2907を介してインターフェース2915と結合されており、一方でD R A M 2910はバス2908を介してインターフェース2915と結合されている。またインターフェース2915は、バス2916を介して、図中で外部デバイス2920として示される1つ以上の外部デバイスと結合されている。外部デバイス2920は、ハードディスク又は光ディスクなどのメモリ、I／Oプロセッサ又は画像処理用プロセッサなどのプロセッサ、トランシーバ（ネットワークトランシーバなど）等を含む複数のデバイスのうちの任意の1つとすることができるが、外部デバイスはこれらに限定されるものではない。バス2916は外部デバイス2920をインターフェース2915と結合するため、1つ以上の通信バスを示し、それらは例えば、パケットネットワーク接続部、交換式ネットワーク接続部、シリアルバス、並列バス、などである。以下に更に説明しているように、バス2908はインターフェース2915を介して、D R A M 2910へのアクセスを直接行う。

30

【0104】

図30を参照すると、まず本発明の概念が、外部メモリ3020に関連して外部デバイスとして例示されている。特に、各A P Uはバス2907を介して、インターフェース2915へ要求を送り、及び／又は、バス2907を介してインターフェース2915から

40

50

の応答を受け取る、1つ以上のチャネルと関連付けられている。これについては図30に例示されており、A PU 2901に対し、チャネル0(要求)(2930)とチャネル1(応答)(2935)により関連付けられている。説明のため、エレメント2930と2935の双方とも、1つ以上のFIFO(first-in-first out)バッファである。簡素化のため、他のA PUにおける、順次転送される要求及び応答の他のチャネルは、図30の破線矢印で示す。ここでは、エレメント2930と2935にはコマンドと状態情報だけが格納されるとする。要求チャネルと応答チャネルはバス2907を介して伝送コントローラ(transfer controller)2945を用いてコマンドと状態情報を交換する。説明のため、伝送コントローラ2945はA PUからの要求の処理を行う外部DMACである。図30からわかるように、インターフェース2915には保護テーブル2940が含まれる。これは、PU2905の内部DMAC(図示せず)において図19と関連して上述した、キー管理テーブル1902に追加するものである。外部保護テーブル(保護テーブル2940など)の利用により、外部メモリ3020から、DRAM2910の1つ以上のサンドボックスへの直接的なデータ・ストリーミングが改善される。説明のため、保護テーブル2940の値は、上述の方法と同様の方法でPUにより割り当てられる。保護テーブル2940を利用して、伝送コントローラ2945は、DRAM2910の1つ、又は複数のサンドボックスと外部メモリ3020との間で、いずれかの方向にデータ通信を行うために、伝送バス2950を制御する。外部保護テーブルは必要とされない点に留意された。しかし、外部保護テーブルがない場合、好適にはPUから、例えば、PUと関連付けられて同様に設けられた要求/応答チャネルから、適切な許可が送られるのが望ましい。上述のように、PUは“信頼できるプログラム(trusted program)”を実行するため、関連の読み出し/書き込み許可の割り当てがなされるべきである。更に、保護テーブル2940が拡張され、外部デバイスへのアクセス、又はその一部へのアクセスを制御することもできる。これについては外部メモリ3020の特定のメモリ・アドレス領域により示される。

[0105]

引き続き図30を参照し、また図31と32を特に参照すると、外部メモリ3020からA PU 2901のローカル・メモリ(図示せず)へのデータ伝送に利用するDLS法が例示されている。この方法は図24について上述した方法の変形であるので、全てのステップを以下に繰り返さず、また図31には全てのステップを示していない。この例では、A PU 2901によってアブレット(前述の、ソフトウエアの種類)が実行されると仮定する。このA PUは外部メモリ3020に現在格納されているデータ上で動作する必要があるものである。まず図31のステップ3105を見ると、PU2905はA PU 2901によって用いられるDRAM2910のメモリの割り当てを行う。更にステップ3110で、PU2905はバス2907を介して、割り当てられたメモリに対する適切なキー保護値をインターフェース2915の伝送コントローラ2945へ与え、DRAM2910のメモリへのアクセス制御を行う保護テーブル2940で用いる。PUはA PU 2901によってアクセス制御が行われるように、そのキー管理テーブルの更新を行うことに加え、PUは上記処理を、図19に関連して上述した方法で行う。ステップ3115で、PU2905はA PU 2901によりアブレットの実行を開始する。このアブレットは現在のところ外部メモリ3020に格納されているデータへのアクセスを要求する。A PU 2901を参照すると、ステップ3130で、このA PUにより、チャネル0を介してインターフェース2915へ要求が送られ、外部メモリ3020などの外部デバイスから、DRAM2910へのデータ伝送が行われる。この要求は、DRAM2910への同期書き込みオペレーションに対するものであり、またこの要求は前述のDMAコマンドと形式が類似している。説明のため、この要求には、例えば外部デバイス(ここでは、外部メモリ3020)や、利用される外部デバイスの一部(ここでは、外部メモリ3020のデータ位置)の識別子、伝送量、DRAM2910の対応アドレス、及び、オペレーションの種類が、読み出しオペレーションか書き込みオペレーションか(ここでは、DRAM2910への同期書き込みオペレーション)、が挙げられる。デバイスや利用されるデバイスの

10

20

30

40

50

一部などの外部デバイス情報やサイズ情報は、例えばアプレット関連情報として与えられる。また、他の種類の外部デバイス・パラメータには、デバイス番号やポート番号、仮想チャネル識別子などが挙げられるが、これらに限定されるものでないことに留意されたい。

【0106】

図32、及び引き続き図30を参照すると、インターフェース2915の伝送コントローラ2915により、ステップ3205において、APU2901からの伝送要求が受け取られる。ステップ3210で、伝送コントローラ2945により保護テーブル2940に格納された値のチェックが行われ、DRAM2910で要求されたメモリの値域が有効であるかが判断される。メモリの値域が無効である場合、ステップ3220で、伝送コントローラ2945により、チャネル1を介した伝送が拒否される。しかし、メモリの値域が有効である場合、ステップ3225で、伝送コントローラにより伝送が開始され、完了するまで伝送バス2950を介して、DMA様式でデータ伝送が継続して行われる(ステップ3230)。伝送が完了すると、伝送コントローラ2945はステップ3235で、応答チャネル1を介して伝送完了メッセージを送る。外部デバイス情報も保護テーブル2940に格納される場合は、ステップ3210や3215などで、APU2901が伝送を進める前に、その外部デバイスが利用許可を有しているかどうかのチェックも行う。

【0107】

本発明によれば、外部メモリ3020からAPU2901へのデータ伝送は、図18に関連して上述しているように、DLSを用いて行われる。これに関連して、図33にはDLSの用途が例示されており、図33ではデータ・フロー(破線矢印により示す)が示されている。同期書き込みオペレーションはインターフェース2915を通じてDRAM2910へ行われ、一方で同期読み出しオペレーションはAPU2901を通じて行われ、ローカル・メモリ2981への格納用に、DRAM2910からデータの読み出しが行われる。図17Bから17Dに関連して上述しているように、DRAM2910による個々の同期書き込みオペレーションと同期読み出しオペレーション処理とが、共用メモリのエンブティ状態、プロッキング状態、及び、フル状態によって制御される(例えば、関連する状態情報のL.Sアドレスフィールド、APU_ID、及びF/Eビット値、により示す)。したがって、例えば、最初にPUがAPUによって用いられるDRAMへのデータ伝送を行うといったことを必要とせずに、外部デバイスからAPUのローカル・メモリへ、効率的に、つまり低オーバーヘッドでデータの伝送を行うことが可能である。換言すれば、PUで実行されるべきI/O処理が、安全かつ同期して、APUにより実行されるアーキテクチャレベルにまで、その実行レベルが引き下げられている。これにより、複数のストリーミングアプリケーションにおいて、APUが、外部メモリの複数のバッファからのデータを素早く読み出し、動作することが可能となる。

【0108】

同期読み出しオペレーション、つまり、DRAM2910から外部デバイス(外部メモリ3020など)への伝送、も上述と同様の方法で可能であることに留意されたい。これについては、図34に例示されており、また引き続き図31、32のフローチャートを参照する。例えば、ステップ3115で、PU2905は、アプレットのAPU2901により実行を開始する。このアプレットは、外部メモリ3020に格納するためにデータの伝送を必要とする。そのため、同期読み出しオペレーションに対する要求が、チャネル0を介してAPUによってインターフェース2915へ送られる(つまり、DRAM2910から外部デバイス(外部メモリ3020)へのデータ伝送)。この要求は前述のDMAコマンドと同様の形式である。この要求には、例えば外部デバイス(ここでは、外部メモリ3020)や、利用される外部デバイスの一部(ここでは、外部メモリ3020のデータ位置)の識別子、伝送量、DRAM2910の対応アドレス、及び、外部デバイスオペレーションが、読み出しオペレーションか書き込みオペレーションか(ここでは、DRAM2910からの同期読み出しオペレーション)、が挙げられる。

【0109】

10

20

30

40

50

50

これに関連して、DLSの利用により、インターフェース2915が同期方式でDRAM2910からのデータの読み出すを可能にし、一方で、APU2901はDRAM2910への書き込みを行う。これについては図3.4に示されており、APU2901のローカル・メモリ2981から外部デバイスへのデータ・フロー（破線矢印により示す）が図3.4に示されている。これに関連して、インターフェース2915がデータの読み出しを試みたかどうかに関わらず、DLSの様々な状態がF/Eビット値により示されている。このため、DRAM2915の関連メモリ・ロケーションに対する上述の状態情報が、図3.5に示すように変更される。この状態情報には、上述のようにF/Eビット3501と、デバイスID3502と、データが伝送される伝送先であるアドレス3503とが含まれる。デバイスID3502により、APUなどのデバイスのタイプが識別され、又はインターフェースなどの別のデバイスが識別される。これに関連して、インターフェース2915にはすでに保護テーブルを持つDMAタイプのコントローラが含まれているので、アドレス・フィールドは要求され得ない点に留意すべきである。つまり、DRAM2910は、同期読み出しオペレーションが許可された場合に、インターフェース2915（デバイスID3502により識別される）へ単に通知を出すだけである。

【0110】

上述のように、図3.0により示されている構成は単に例示的なものであり、その他のタイプの外部デバイスへ直接的に拡張することができる。例えば、図3.6を検討する。この図は図3.0と同じものである。このため同じ参考符号は同様の要素を表しており、そのような要素についてはここではさらに説明しない。図3.6では、外部デバイスは物理レシーバ（physical receiver：PHY）3620によって示される。物理レシーバ3620はバス3621を介して、イーサネットなどのネットワーク（図示せず）からパケット通信を受信し、受信したビット・ストリームのデコーディングを行い、パケットのストリームを与える。この例では、APUによって実行されたアプレットは受信したパケットストリームのリンク層処理を示す。つまり、APU2901はPHY3620からのデータの処理を行うために上述のDLS機能を利用して、I部のプロトコルスタック（TCP/IP（通信制御プロトコル／インターネットプロトコル）など）に対し処理を行う。

【0111】

引き続き図3.6を参照し、また特に図3.7を参照すると、例示的なハンドシェーティング・シーケンスが示されており、外部デバイスを用いたデータレベル同期の利用がさらに例示されている。この図では、共有メモリ（DRAM2910など）の対応部分がエンプティ状態で開始されると仮定する。最初に、APU（APU2901など）が要求チャネルを介して、外部デバイスから共用メモリの識別部分への同期書き込みオペレーションを要求するコマンドを、インターフェース（インターフェース2915など）へ送る。ここでは、バスインターフェース2915のチェックを行う保護テーブルが、外部デバイスを用いて伝送を開始すると仮定する。一方で、APUは共用メモリのこれらの部分からの同期読み出しオペレーションの実行を試みる。インターフェースによって書き込みが行われているデータはまだないが、共用メモリの対応部分がブロッキング状態に入る。この状態で、前述のように、APUに対するローカル・メモリのアドレスが共用メモリのこれらの部分に対し、関連する状態情報を格納される。続いて、インターフェースが共用メモリの識別部分への同期書き込みオペレーションを行った後、APUに対し、同期読み出しオペレーションが、格納されたアドレス値により示されるアドレスへ行われる。その結果、共用メモリのこれらの部分がエンプティ状態へ戻る。その他の順列に対する状態変化（図1.8に示す）を示す、同様のハンドシェーティング・シーケンスは簡明であり、ここには説明していない。

【0112】

1つ以上のAPUからの要求は、インターフェースを通じて、各要求の完了を待つ必要がないことがわかる。例えば、APUは1つ以上の外部デバイスからのデータに対し、インターフェースへ複数の要求を出すことができる。DLSの利用により、（インターフェースを介して）APU、あるいは外部デバイスのいずれから、同期読み出しオペレーション

10

20

30

40

50

ンが行われるまで、データが確実に共用メモリに保護される。そのため、DLSの利用により、APUと1つ以上の外部デバイスとの間に実行されるバースト伝送と並列バースト伝送とが可能になる。

【0113】

説明のため、図36に示すデバイスを用いた本発明の概念の他の適用例を図38に示す。APU2901はインターフェース2915を介して、多数の外部デバイスと接続している。これらの外部デバイスは、USB(ユニバーサル・シリアル・バス)デバイス、IEEE1394(一般に“ファイアワイヤ”と呼ばれる)デバイス、シリアル・デバイス、ハードディスク、ネットワーク・デバイス(ネットワークトランシーバなど)、光ディスク、である。その他の外部デバイス(図示せず)としては、I/Oプロセッサ、又は画像処理用プロセッサなどのプロセッサが挙げられる。APU2901は1つ以上のこれらの外部デバイスからのデータの読み出しを行うために、要求4301を送る。そのため、各デバイスはインターフェース2915の制御を介して、統いてDRAM2910のそれぞれに割り当てられたメモリ・ロケーションへ同期書き込みオペレーションを行う。DRAM2910のメモリ・ロケーションは、メモリ・ロケーション4311、4312、4313、4314、4315、4316により示される(各メモリ・ロケーションには、関連付けられた状態情報が含まれることも示し、それらは単にF/Eビットにより示される)。APU2901は、同期読み出しオペレーションを行うことにより、バス2906を介して、DRAM2910からデータを読出す。

【0114】

図39を参照すると、本発明の概念の別の実施形態が例示されている。図39に簡略化したストリーミング構成3700を示す。構成3700には、PE3790と、DRAM(又は共用メモリ)3710と、インターフェースデバイス(インターフェース)3715と、物理トランシーバ(PHY)により示される外部デバイスとが含まれる。外部デバイス3720はインターフェース3715の一部ともでき、例えば、インターフェース3715と結合することもできる。PE3790は1つ以上の処理エレメントを示し、更にPE3790には少なくとも1つのPUS3705と、1つ以上のAPU(APU3701、3702、3703により示す)とが含まれる。PE3790はバス3706を介して共用メモリと結合されている。PE3790はまた、前述のDLS機構とメモリ保護機構を用いて、DRAM3710へのデータの書き込みと、DRAM3710からのデータの読み出しを行う。本発明の概念によれば、現在のところ、DLS機構は外部デバイスと用いるように拡張されている。そのため、また、図39からわかるように、各APUは、バス3707を介してインターフェース3715と結合されている。特に各APUは、1つ以上のチャネルと関連付けられており、インターフェース3715への要求の送信、及び/又は、インターフェース3715からの応答の受信、及び/又は、インターフェース3715とのデータ通信が、バス3707を介して行われる。これについては図39に例示されており、APU3701に対して、チャネル0(入力)(3730)と、チャネル1(出力)(3735)が関連付けられている。説明のため、エレメント3730と3735の双方が、1つ、またはそれ以上のFIFOバッファである。その他のAPUに関しては、バス3707を介して転送されるそれらの入力及び出力のチャネルは、図39に破線矢印により示される。

【0115】

インターフェース3715には、チャネル・インターフェース・エレメント3745と3750とが含まれ、インターフェース3715により、APU3701とPHY3720との間のそれぞれの出入力データストリームが結合される。説明のため、チャネル・インターフェース・エレメント3750はトランシーバであり、(a)APU3701の出力チャネルから送られるデータを、バス3707と3716を介してエンコードし、バス3721を介して伝送する、及び(b)バス3721から受け取ったデータをデコードし、バス3716と3707を介して、APU3701の入力チャネルへデコードしたデータを送る。バス3721は外部デバイス3720を、パケットネットワーク接続部、交換

10

20

30

40

50

式ネットワーク接続部、シリアルバス、並列バス、などへ結合するための、1つ以上の通信バスを示す。

【0116】

上述のように、インターフェース3715には、チャネル・インターフェース・エレメント3745と3750が含まれる。チャネル・インターフェース・エレメント3745はAPU3701から出力チャネルを受信する。本発明によれば、出力チャネル自体には制御チャネルとデータチャネルとが含まれる。この制御チャネルにより、上の要求と応答のチャネルが形成され、一方で、データのチャネルにより、外部デバイスへ送られるデータが転送される。この例では、要求チャネルにより外部デバイス識別子（ここではPHY3720）を含むコマンド情報や、実行されるオペレーション（ここでは、書き込みオペレーション）などが転送される。出力チャネルのデータチャネル部は、PHY3720によってエンコードされ、バス3721を介して伝送されるデータを示す。同様に、チャネル・インターフェース・エレメント3745は、APU3701の入力チャネルへ結合される。入力チャネル自体には、制御チャネルとデータチャネルとが含まれる。制御チャネルにより、上の要求と応答のチャネルが形成され、一方で、APUへ送られるデータはデータチャネルによって転送される。この例では、外部デバイス識別子（ここでは、PHY3720）や、実行されるオペレーション（ここでは、読み出しオペレーション）などを含むコマンド情報が、要求チャネルによって転送される。この入力チャネルのデータチャネル部により、バス3721から受信した信号受信の結果として、PHY3720によってデコードされたデータが示される。

10

【0117】

この実施形態では、インターフェース3715内には保護テーブルが必要とされないことがわかる。特に、また更に以下に説明しているように、図18に例示している前述のDLS機構とメモリ保護機構により、各APUはDRAM3710からのデータの読み出しどと、DRAM3710へのデータの書き込みどとが行われる。

【0118】

引き続き図39を参照すると、外部デバイスからAPU2901のローカル・メモリ（図示せず）へのデータ伝送に用いる例示的な方法が図40に示されている。この方法により、上述した図24の方法も実行されると仮定する。その方法とは、P_Uによる保護テーブル値の割り当て、APU3701などへのアプレットの割り当て、などである。説明のため、この例では、APU3701によって実行されるアプレットは、受信したパケットストリームのリンク層処理と、送信したパケットストリームに対するリンク層処理とを示す。つまり、APU2901によって上述のDLS機構を利用したプロトコルスタック（TCP/IPなど）の一部分に対する処理が行われる。

20

【0119】

図40のステップ3830をまず参照すると、APU3701は、バス3707を介して、PHY3720からの、また、PHY3720へのデータ伝送要求を、インターフェース3715へ行う。この例では、入力チャネルと出力チャネルの双方のコマンド部を介してこの要求が行われる。しかし、入力チャネル及び出力チャネルの双方の利用は必要とされない。この要求には、外部デバイス（ここではPHY3720）の識別子などが含まれ、また、利用する外部デバイスの一部（特定のポートなど）や、外部デバイスのオペレーションの種類（ここでは、入力チャネルに対する外部デバイスからの読み出しどと、出力チャネルに対する外部デバイスへの書き込み）も含まれる。デバイスや利用するデバイスの一部などの外部デバイス情報は、アプレット間連の情報などに与えられるとする。さらに、他の種類の外部デバイスパラメータには、デバイス番号やポート番号、仮想チャネル識別子、などを挙げられるが、これらに限定されるものではない。

30

【0120】

ステップ3805で、インターフェース3715は要求を受け取り、PHY3720への、またPHY3720からの伝送を開始する。特定のハンドシェーティングがPHY3720に必要とされる範囲で、このハンドシェーティングがインターフェース3715によっ

40

50

て行われ、また、これが APU 3701 にとってトランスペレントなものであるとする。ステップ 3710 で、インターフェース 3715 は伝送を開始し、伝送が完了するまで、PHY 3720 からのデータ伝送を継続する（ステップ 3815）。このようなデータ伝送は、例えば遠方終端への接続が切れるまでは継続される。この、PHY 3720 への、また PHY 3720 からのデータ伝送の部分は、外部デバイスの種類に応じて、同期して、又は非同期に行うことができる。データは入力チャネルのデータチャネル部内に、例えば、エレメント 3735 の FIFO の一部などへ、伝送される。伝送が完了すると、インターフェース 3715 はステップ 3820 で、入出力チャネルのコマンド部の応答チャネルを介して、伝送完了メッセージを送る。

【0121】

本発明によれば、APU 3701 による、外部デバイス 3720 からの、また、外部デバイス 3720 へのデータ伝送は、DLS を用いて行われる。この DLS の状態図は先述の図 18 に説明している。これに関連して、DLS の用途を図 41 に例示する。APU と外部デバイスとの間のデータ・フローが破線矢印により示されている。図 41 からわかるように、APU 3701 はバス 3707 による入力チャネルを介して、上述しているようにローカル・メモリの中にデータを直接受取る。同様に、APU 3701 はバス 3707 による出力チャネルを介して、ローカル・メモリから直接データを送る。これにより、速くて低いオーバーヘッドの通信チャネルがもたらされる。例えば、PHY 3720 を介して送信されるデータは、まず DRAM 3710 に格納され得る。APU 3701 は同期読み出しオペレーションを介して、DLS を用いて、このデータを DRAM 3710 から抽出する。これについては図 41 に対応の破線矢印により例示している。次いで、APU 3701 はリンク層のアプレットによりこのデータの処理を行い、処理したデータ（フォーマットされたパケットのストリームなど）を（インターフェース 3715 を介して）PHY 3720 へ送り、更にエンコードを行い、伝送する。同様に、PHY 3720 によって受け取られたデータのデコードが行われ、データはローカル・メモリ 3781 に格納するために入力チャネルを介して APU 3701 へ送られ、APU 3701 によって処理される。APU 3701 はリンク層のアプレット（ヘッダの除去、エラーチェックなど）によりデータの処理を行い、同期書き込みオペレーションを介して、DLS を用いて、処理されたデータを DRAM 3710 へ送る。これについては図 41 に、対応の破線矢印で描かれている。

【0122】

図 42 に図 39 に関連して示され、上記に説明されている、簡略化したストリーミング構成の変形を例示する。この実施形態は図 39 に示す方法と同様の方法で動作するため、全ての図については詳細に説明していない。図 42 の構成 4000 には PE 4090 と、DRAM（又は、共用メモリ）4010 と、インターフェース 4015 と、物理トランシーバ（PHY）4020 と物理トランシーバ（PHY）4080 により示される外部デバイス、とが含まれる。図 42 からわかるように、説明のため、インターフェース 4015 は外部デバイス 4020 及び 4080 と結合されている。PE 4090 は 1 つ以上の処理エレメントを示し、更に、PE 4090 には少なくとも 1 つの PU 4005 と、1 つ以上の APU（APU 4001、4002、4003）とが含まれる。PE 4090 はバス 4006 を介して共用メモリと結合されており、前述の DLS とメモリ保護機構を用いて、DRAM 4010 へのデータの書き込みと、DRAM 4010 からのデータの読み出しを行なう。図 42 からわかるように、APU はバス 4007 を介して、インターフェース 4015 と結合されている。特に、各 APU は 1 つ以上のチャネルと関連付けられており、バス 4007 を介して、インターフェース 4015 への要求の送信、及び／又は、インターフェース 4015 からの応答の受信、及び／又はインターフェース 4015 とのデータ通信を行う。これについては図 42 に例示されており、APU 4001 に対し、入力／出力チャネル 4050 と入力／出力チャネル 4055 が関連付けられている。説明のため、これらのエレメントは双方とも、1 つ以上の FIFO バッファである。バス 4007 を介して形成されるその他の APU 用のその他の入力及び出力のチャネルは、図 42 の破線矢

10

20

30

40

50

印により示されている。

【0123】

インターフェース4015にはエレメント4030と4035とが含まれ、各エレメントにはそれぞれ、2つのチャネル・インターフェース・エレメント4031、4032と4036、4037とが含まれる。(これらは図39で説明している種類である。)これらのエレメントにより、データストリームがそれぞれの外部デバイス(ここでは、4020と4080)へ結合される。説明のため、この外部デバイスは、それぞれのバス4021と4081への、及び、それぞれのバス4021と4081からのデータ通信用トランシーバである。この外部デバイス4020、及び／又は4080には、例えばハードディスク又は光ディスクなどのメモリ、1／0プロセッサ又は画像処理用プロセッサなどのプロセッサ、トランシーバ(ネットワークトランシーバなど)なども挙げられる。

10

【0124】

上述の通り、この例では、APU4001は2つの入力／出力チャネル(4050と4055)へのアクセスを持ち、各チャネルには更に、コマンド情報とデータを転送する、上述の入出力チャネルが含まれる。上述した方法と同様の方法で、APU4001によってアブレットが実行され、外部デバイス(4020と4080)との間にデータが転送される。APU4001は複数の入力／出力チャネルへのアクセスを持つので、APU4001がDRAM4010へアクセスすることは要求されない。そのために、図42に示す変形は、外部デバイス間にデータを転送する、効率的で高速な構成を提供する。

20

【0125】

このように、上記の内容は、単に本発明の原則を例示するものに過ぎず、当業者は、ここには明示的に記載されていないが、本発明の趣旨及び範囲から逸脱することなくその原則を具体化する数多くの代替の構成を考案しうるであろうことを理解されよう。例えば、本発明の概念が、データレベルの同期化を行う入力／出力インターフェースを探り上げて説明されているが、インターフェースが必ずしも入力と出力の双方を行う必要はなく、例えば、インターフェースは入力デバイスのみ、又は、出力デバイスのみ、又は、一般的の入力／出力ネットワークで用いられる、入力デバイスと出力デバイスの任意の組合せ、などのうちの1つでありうる。

【図面の簡単な説明】

【0126】

【図1】本発明によるコンピュータ・ネットワークのアーキテクチャ全体を示す説明図。

30

【図2】本発明によるプロセッサ・エレメント(P.E.)の構造を示す説明図。

【図3】本発明による広帯域エンジン(B.E.)の構造を示す説明図。

【図4】本発明による付加処理デバイス(APU)の構造を示す説明図。

【図5】本発明によるプロセッサ・エレメントと、ビジュアライザ(V.S.)と、光インターフェースとの構造を示す説明図。

【図6】本発明によるプロセッサ・エレメントの1つの組合せを示す説明図。

【図7】本発明によるプロセッサ・エレメントの別の組合せを示す説明図。

【図8】本発明によるプロセッサ・エレメントの更に別の組合せを示す説明図。

【図9】本発明によるプロセッサ・エレメントの更に別の組合せを示す説明図。

40

【図10】本発明によるプロセッサ・エレメントの更に別の組合せを示す説明図。

【図11A】本発明によるチップ・パッケージ内へ光インターフェースを統合した例を示す説明図。

【図11B】図11Aの光インターフェースを用いるプロセッサの1つの構成を示す説明図。

【図11C】図11Aの光インターフェースを用いるプロセッサの別の構成を示す説明図。

【図12A】本発明によるメモリ・システムの構造を示す説明図。

【図12B】本発明による第1の広帯域エンジンから第2の広帯域エンジンへのデータの書き込みを示す説明図。

50

【図 1 3】本発明によるプロセッサ・エレメント用の共用メモリの構造を示す説明図。

【図 1 4 A】図 1 3に示すメモリ・バンクの一構造を示す説明図。

【図 1 4 B】図 1 3に示すメモリ・バンクの別の構造を示す説明図。

【図 1 5】本発明による DMA C の構造を示す説明図。

【図 1 6】本発明による DMA C の代替の構造を示す説明図。

【図 1 7 A】本発明によるデータ同期オペレーションを示す説明図。

【図 1 7 B】本発明によるデータ同期オペレーションを示す説明図。

【図 1 7 C】本発明によるデータ同期オペレーションを示す説明図。

【図 1 7 D】本発明によるデータ同期オペレーションを示す説明図。

【図 1 7 E】本発明によるデータ同期オペレーションを示す説明図。

【図 1 7 F】本発明によるデータ同期オペレーションを示す説明図。

【図 1 7 G】本発明によるデータ同期オペレーションを示す説明図。

【図 1 7 H】本発明によるデータ同期オペレーションを示す説明図。

【図 1 7 I】本発明によるデータ同期オペレーションを示す説明図。

【図 1 7 J】本発明によるデータ同期オペレーションを示す説明図。

【図 1 7 K】本発明によるデータ同期オペレーションを示す説明図。

【図 1 7 L】本発明によるデータ同期オペレーションを示す説明図。

【図 1 7 M】本発明によるデータ同期オペレーションを示す説明図。

【図 1 7 N】本発明によるデータ同期オペレーションを示す説明図。

【図 1 7 O】本発明によるデータ同期オペレーションを示す説明図。

【図 1 8】本発明によるデータ同期方式によるメモリ・ロケーションの様々な状態を示す説明図。

【図 1 9】本発明によるハードウェア・サンドボックス用のキー管理テーブルの構造を示す説明図。

【図 2 0】本発明によるハードウェア・サンドボックス用のメモリ・アクセス・キーの格納方式を示す説明図。

【図 2 1】本発明によるハードウェア・サンドボックス用のメモリ・アクセス管理テーブルの構造を示す説明図。

【図 2 2】図 1 9 のキー管理テーブルと、図 2 1 のメモリ・アクセス制御テーブルとを用いてメモリ・サンドボックスにアクセスするステップを示すフローチャート。

【図 2 3】本発明によるソフトウェア・セルの構造を示す説明図。

【図 2 4】本発明による、APUへ過隔処理命令を出すステップを示すフローチャート。

【図 2 5】本発明によるストリーミング・データ処理専用パイプラインの構造を示す説明図。

【図 2 6 A】本発明によるストリーミング・データの処理時の図 2 5 の専用パイプラインによって実行されるステップを示すフローチャート。

【図 2 6 B】本発明によるストリーミング・データの処理時の図 2 5 の専用パイプラインによって実行されるステップを示すフローチャート。

【図 2 7】本発明によるストリーミング・データ処理専用パイプラインの他の構造を示す説明図。

【図 2 8】本発明によるAPUによるアプリケーションとデータの並列処理を調整するための絶対タイマー方式を示す説明図。

【図 2 9】本発明の原則に従って、データレベル同期を用いた構成の実施形態を例示的に示す説明図。

【図 3 0】本発明の原則に従って、データレベル同期を用いた構成の実施形態を例示的に示す説明図。

【図 3 1】本発明の原則に従って、データレベル同期実行時に用いるフローチャートを例示的に示す説明図。

【図 3 2】本発明の原則に従って、データレベル同期実行時に用いるフローチャートを例示的に示す説明図。

10

20

30

40

50

【図 3 3】 A P U と外部デバイスとの間にデータレベル同期の利用を示す説明図。

【図 3 4】 A P U と外部デバイスとの間にデータレベル同期の利用を示す説明図。

【図 3 5】 本発明の原則に従ってデータレベル同期状態情報を例示的に示す説明図。

【図 3 6】 本発明の原則に従ってデータレベル同期を用いた構成のもう 1 つの実記形態を例示的示す説明図。

【図 3 7】 ハンドシェーディングシーケンスを例示的に示す説明図。

【図 3 8】 本発明の原則に従って、データレベル同期を用いた構造用の別の実施形態を例示的に示す説明図。

【図 3 9】 本発明の原則に従い、データレベル同期実行時に用いるフローチャートを例示的に示す図。

【図 4 0】 本発明の原則に従い、データレベル同期の実行に用いる別のフローチャートを例示的に示す説明図。

【図 4 1】 A P U と外部デバイスとの間に別のデータレベル同期の用途を例示的に示す説明図。

【図 4 2】 本発明の原則に従いデータレベル同期を用いた別の構成用の実施形態を例示的に示す説明図。

【符号の説明】

【0 1 2 7】

1 0 1 システム

1 0 1 0 キー

10

1 0 2 セル

1 0 4 ネットワーク

20

1 0 6 クライアント

1 0 8 サーバー・コンピュータ

1 1 0 4 光インターフェース

1 1 0 8 バス

1 1 1 8, 1 1 2 2 ポート

1 1 2 6 光導波路

1 1 6 0, 1 1 6 2, 1 1 6 4, 1 1 6 6, 1 1 8 2, 1 1 8 4, 1 1 8 6, 1 1 8 8,

30

1 1 9 0 光インターフェース

1 2 0 6, 1 2 3 4, 1 2 4 2 コントロール

1 2 1 2, 1 2 4 0 ユニット

1 2 2 1 クロスバ交換機

1 2 3 2 外部ポート

1 2 4 4, 1 4 1 4, 1 4 1 6 バンク

1 4 0 6 ブロック

40

1 5 0 4 ノード

1 6 0 7, 1 6 0 8 バス

1 7 2 2 制御回路

1 7 2 4, 1 7 4 2 制御論理回路

1 7 2 6 ストレージ

1 7 2 8, 1 7 3 1, 1 7 3 2, 1 7 4 6, 1 7 5 0 ロケーション

1 7 2 9, 1 7 5 2, 1 7 6 0, 1 7 6 2 セグメント

1 8 8 0 エンブディ状態

1 8 8 2 フル状態

1 8 8 4 ブロッキング状態

1 9 0 2 キー管理テーブル

1 9 0 6 キー

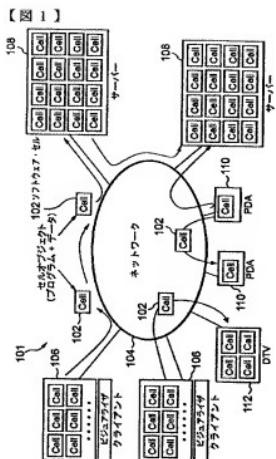
1 9 0 8 マスク

2 0 0 6 格納位置

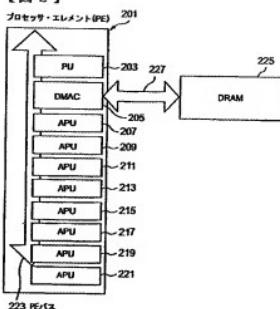
50

2 0 0 8, 2 0 1 0	セグメント	
2 0 1 2	キー	
2 1 0 2	アクセス管理テーブル	
2 1 0 6	アドレス	
2 1 1 0	キー	
2 1 1 0	キー・マスク	
2 2 3	バス	
2 2 7	高帯域メモリ接続部	
2 3 0 2, 2 3 2 0	セル	10
2 3 0 8	ヘッダ	
2 3 2 2	インターフェース	
2 3 3 2	実行セクション	
2 3 3 4	リスト	
2 5 2 0, 2 5 2 2, 2 5 2 4, 2 5 2 6, 2 7 0 4	サンドボックス	
2 7 0 6	宛先サンドボックス	
2 9 0 0	処理環境	
2 9 0 1, 2 9 0 2, 2 9 0 3	APU	
2 9 0 5	P U	
2 9 0 6～2 9 0 8	バス	
2 9 1 0	D R A M	20
2 9 1 5	インターフェース	
2 9 2 0	外部デバイス	
2 9 3 0	エレメント	
2 9 4 0	保護テーブル	
2 9 4 5	伝送コントローラ	
2 9 5 0	伝送バス	
2 9 8 1	メモリ	
3 0 2 0	外部メモリ	
3 6 2 0	物理レシーバ	
3 7 0 6, 3 7 0 7	バス	30
3 7 1 5	インターフェース	
3 7 2 0	外部デバイス	
3 7 3 0, 3 7 3 5, 3 7 4 5, 3 7 4 0	エレメント	
3 7 8 1	メモリ	
4 0 1 5	インターフェース	
4 0 2 0	外部デバイス	
4 0 3 0, 4 0 3 1	エレメント	
4 0 5 0, 4 0 5 5	出力チャネル	

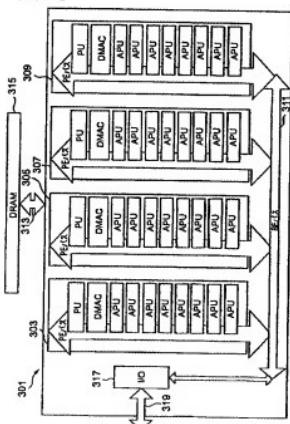
【図1】



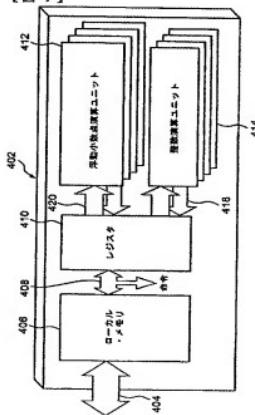
【図2】



[图 3]



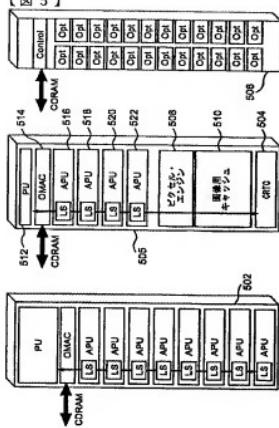
[図 4]



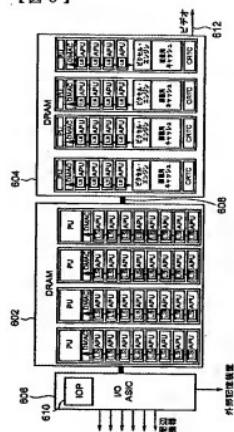
(37)

JP 2006-107514 A 2006.4.20

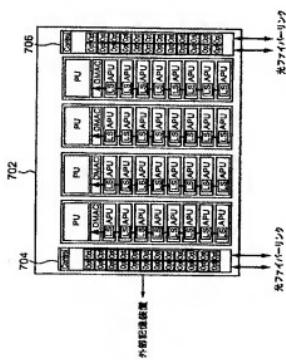
【図5】



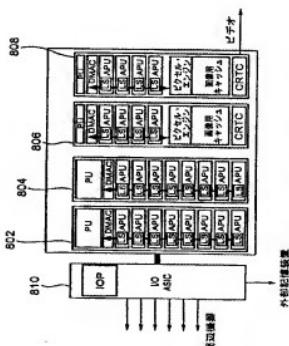
【図6】



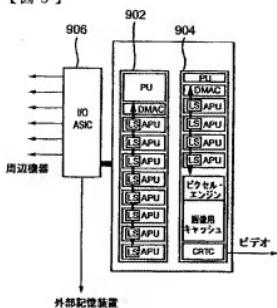
【図7】



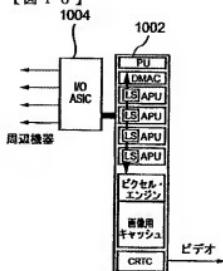
【図8】



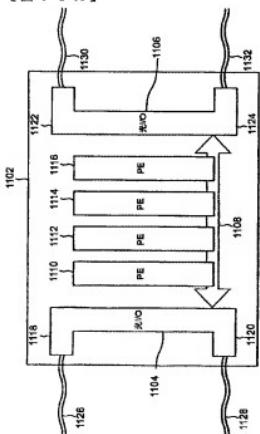
【図9】



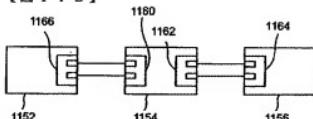
【図 10



【図 1-1A】

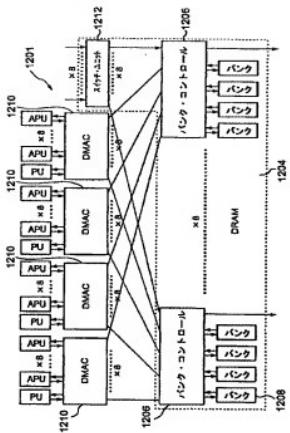


[图 11B]

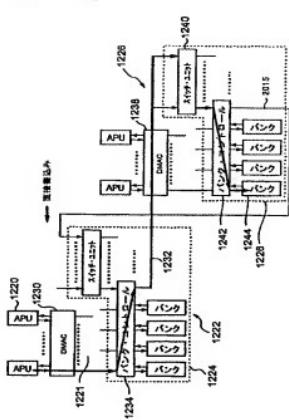


[图 1-1c]

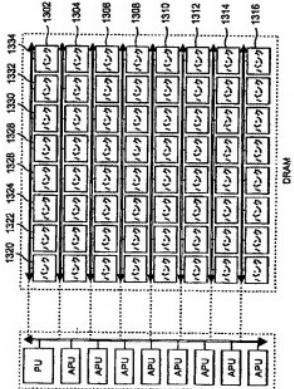
【図12A】



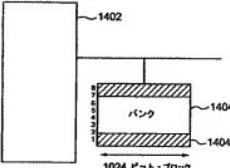
【図12B】



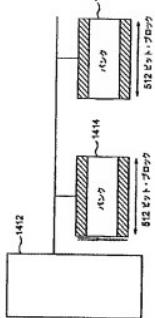
【図13】



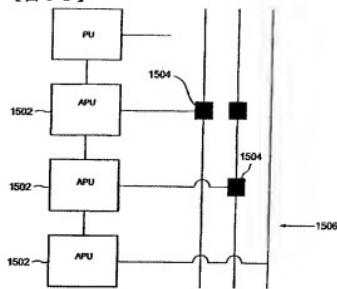
【図14A】



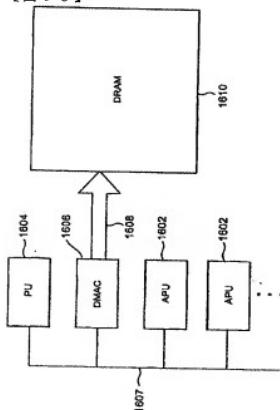
【図14B】



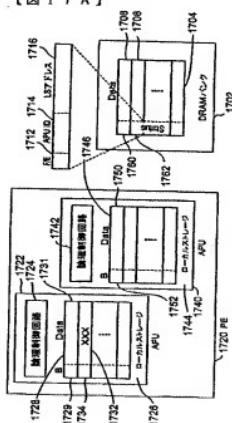
【図15】



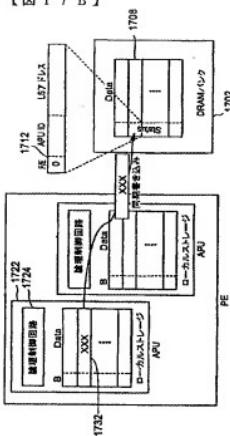
【図16】



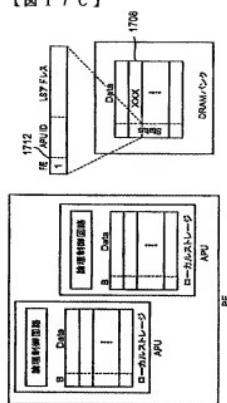
【図17A】



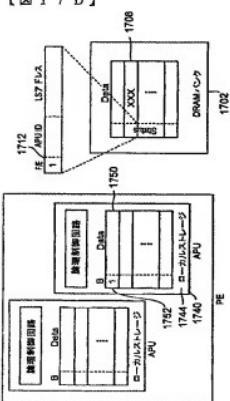
【図 17 B】



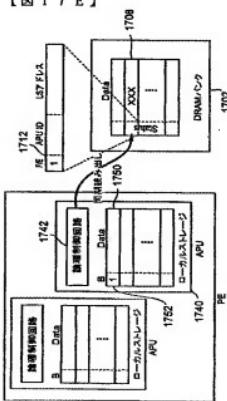
【図 17 C】



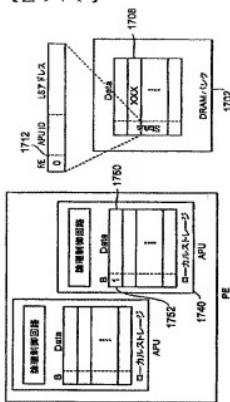
【図 17 D】



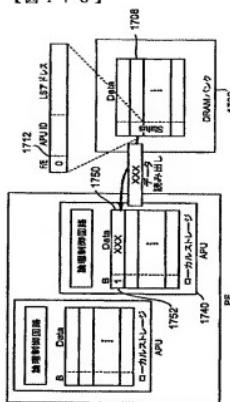
【図 17 E】



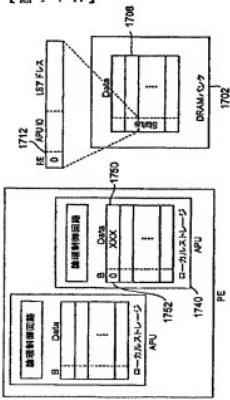
【図17F】



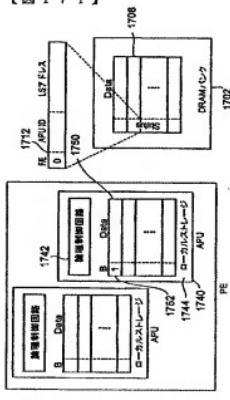
【図17G】



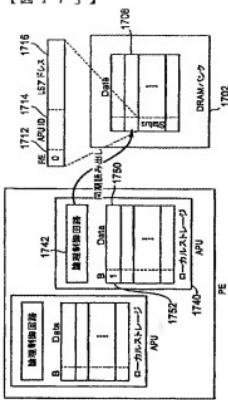
【図17H】



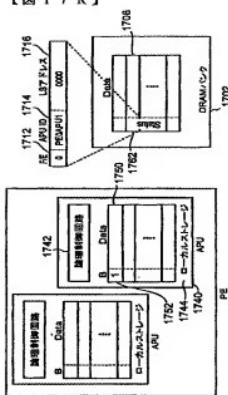
【図17I】



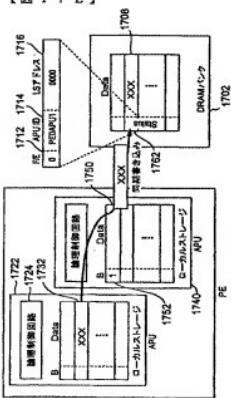
【図 17 J】



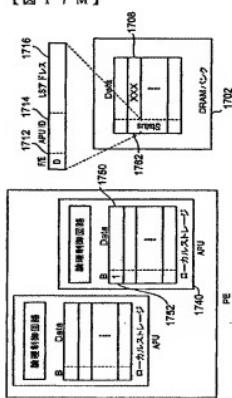
【図 17 K】



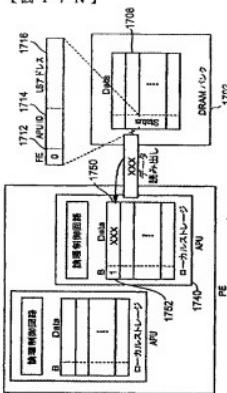
【図 17 L】



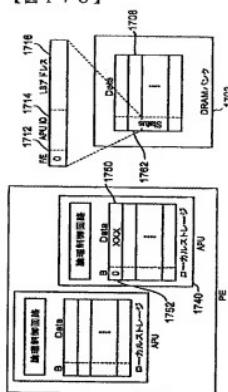
【図 17 M】



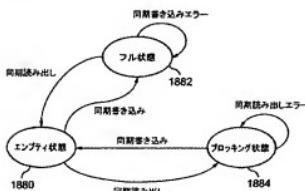
【図17N】



【図17O】



【図18】

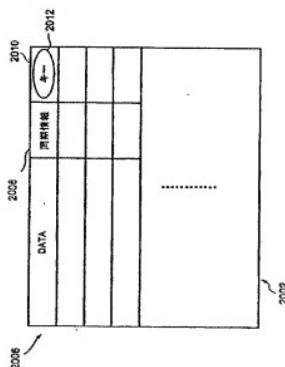


【図19】

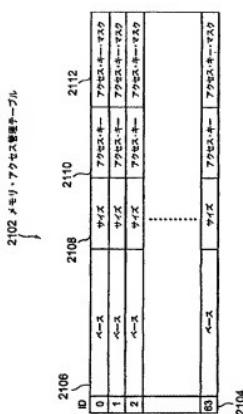
キー管理テーブル 1902

ID	APUキー	キー・マスク
0	APUキー	キー・マスク
1	APUキー	キー・マスク
2	APUキー	キー・マスク
⋮	⋮	⋮
7	APUキー	キー・マスク

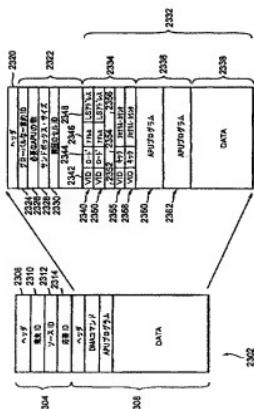
【図20】



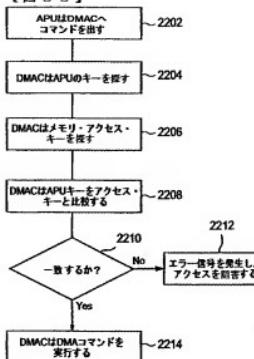
【図21】



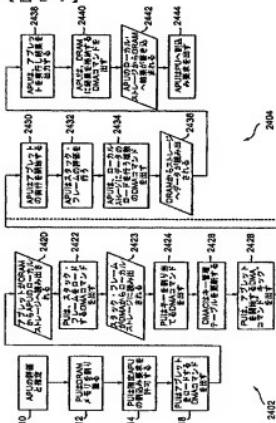
〔圖23〕



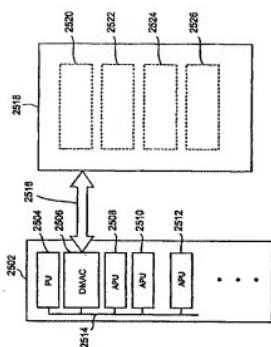
【图 22】



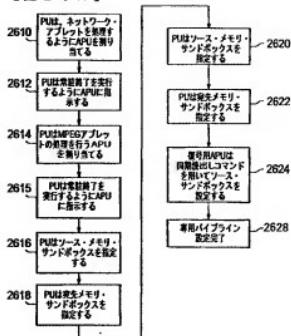
【図24】



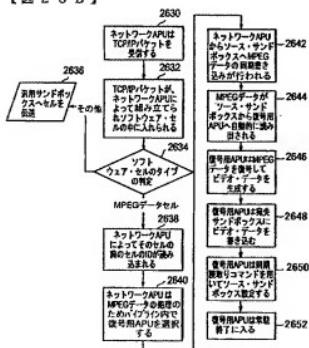
【図25】



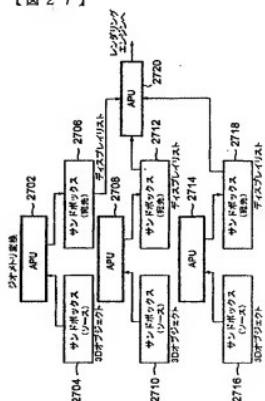
【図26A】



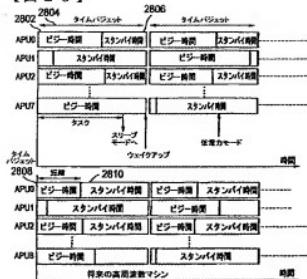
【図26B】



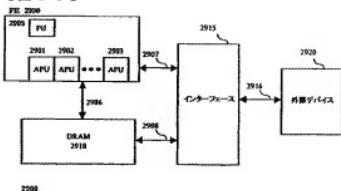
【図27】



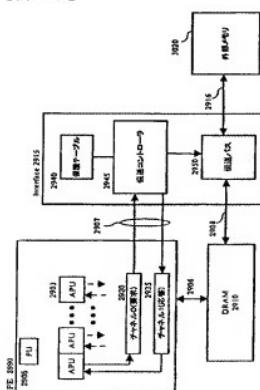
【図28】



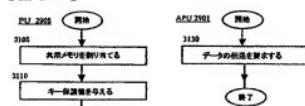
【图29】



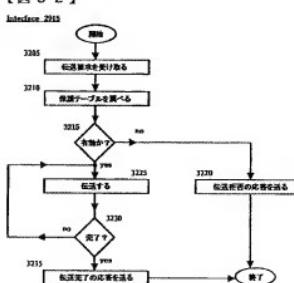
〔図30〕



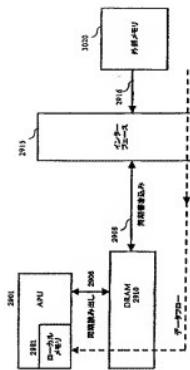
[图31]



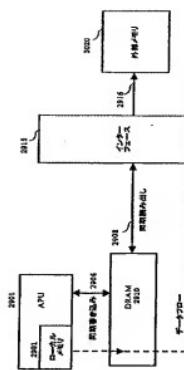
[图32]



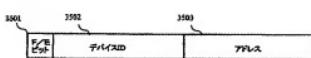
【图 3-3】



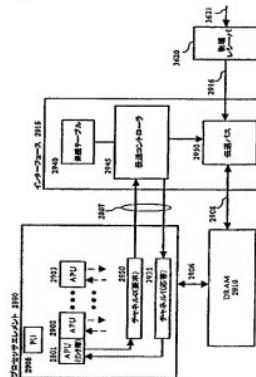
【图 3-4】



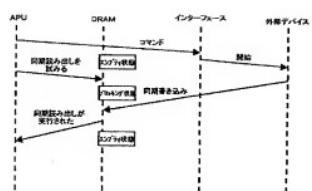
【圖 35】



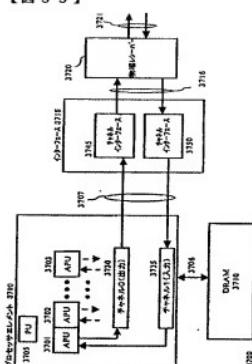
【图 3-6】



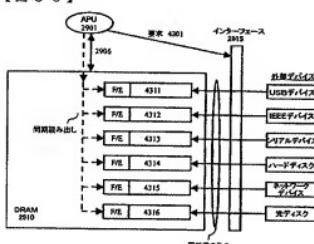
【図37】



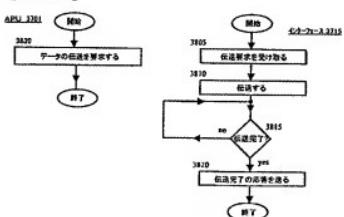
【図39】



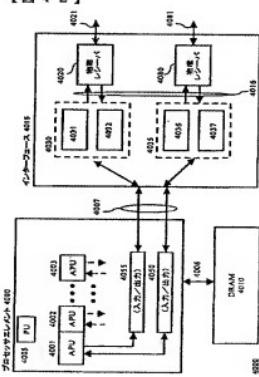
【図38】



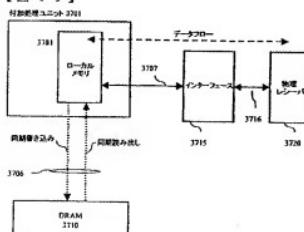
【図40】



【図42】



【図41】



フロントページの続き

(72)発明者 山崎 剛

東京都港区南青山二丁目6番21号 株式会社ソニー・コンピュータエンタテインメント内